

DISSERTATION

APPLYING MODEL-BASED SYSTEMS ENGINEERING TO ARCHITECTURE  
OPTIMIZATION AND SELECTION DURING SYSTEM ACQUISITION

Submitted by

Michael LaSorda

Walter Scott, Jr. College of Engineering

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Fall 2018

Doctoral Committee:

Advisor: Ronald M. Sega

Co-Advisor: Mike Boriky

Tom Bradley

Jason Quinn

Copyright by Michael J. LaSorda 2018

All Rights Reserved

## ABSTRACT

### APPLYING MODEL-BASED SYSTEMS ENGINEERING TO ARCHITECTURE OPTIMIZATION AND SELECTION DURING SYSTEM ACQUISITION

The architecture selection process early in a major system acquisition is a critical step in determining the overall affordability and technical performance success of a program. There are recognized deficiencies that frequently occur in this step such as poor transparency into the final selection decision and excessive focus on lowest cost, which is not necessarily the best value for all of the stakeholders. This research investigates improvements to the architecture selection process by integrating Model-Based Systems Engineering (MBSE) techniques, enforcing rigorous, quantitative evaluation metrics with a corresponding understanding of uncertainties, and stakeholder feedback in order to generate an architecture that is more optimized and trusted to provide better value for the stakeholders. Three case studies were analyzed to demonstrate this proposed process. The first focused on a satellite communications System of Systems (SoS) acquisition to demonstrate the overall feasibility and applicability of the process. The second investigated an electro-optical remote sensing satellite system to compare this proposed process to a current architecture selection process typified by the United States Department of Defense (U.S. DoD) Analysis of Alternatives (AoA). The third case study analyzed the evaluation of a service-oriented architecture (SOA) providing satellite command and control with cyber security protections in order to demonstrate rigorous accounting of uncertainty through the architecture evaluation and selection. These case studies serve to define and demonstrate a new, more transparent and trusted architecture selection process that consistently provides better value for the stakeholders of a major system acquisition. While the examples in this research focused on U.S.

DoD and other major acquisitions, the methodology developed is broadly applicable to other domains where this is a need for optimization of enterprise architectures as the basis for effective system acquisition. The results from the three case studies showed the new process outperformed the current methodology for conducting architecture evaluations in nearly all criteria considered and in particular selects architectures of better value, provides greater visibility into the actual decision making, and improves trust in the decision through a robust understanding of uncertainty. The primary contribution of this research then is improved information support to an architecture selection in the early phases of a system acquisition program. The proposed methodology presents a decision authority with an integrated assessment of each alternative, traceable to the concerns of the system's stakeholders, and thus enables a more informed and objective selection of the preferred alternative.

It is recommended that the methodology proposed in this work is considered for future architecture evaluations.

## ACKNOWLEDGEMENTS

The author would like to thank everybody who contributed to the creation of this work. Of note were Mr. John Morris, Military Satellite Communications Systems Directorate Chief Engineer at the U.S. Air Force Space and Missile Systems Center who provided the initial inspiration for this work and Mr. Frederic Agardy, Satellite Communications Chief Architect for the Aerospace Corporation, who provided valuable guidance. The author would also like to specifically thank the members of the committee and the staff of the systems engineering program at Colorado State University for their helpfulness, patience and excellent responsiveness.

In particular the author would like to recognize and thank Dr. Mike Borky for his invaluable mentorship and encouragement over the course of this research, without which I would have given up long ago. It was truly a stroke of luck to be matched with Dr. Borky on this research.

## TABLE OF CONTENTS

ABSTRACT.....	ii
ACKNOWLEDGEMENTS.....	iv
TABLE OF CONTENTS.....	v
LIST OF TABLES.....	viii
LIST OF FIGURES.....	ix
Chapter 1: Introduction.....	1
1.1 Content of the Dissertation.....	1
1.2 Problem Overview.....	2
1.3 Literature Review.....	4
1.3.1 MBSE Overview.....	4
1.3.2 Optimization Techniques in Engineering.....	7
1.3.3 Applications of Optimization to System Design.....	8
1.3.4 MBSE and Optimization Integration.....	11
1.3.5 Uncertainty Analysis in Optimization.....	13
1.4 Proposed Solution.....	18
Chapter 2: Overview of Approach.....	21
2.1 Reference Architecture Generation.....	21
2.1.1 Reference Architecture Overview.....	21
2.1.2 SysML Introduction.....	22
2.2.3 Reference Architecture Organization.....	29
2.2 Contributing Analyses Selection.....	30
2.2.1 Criteria for Contributing Analyses.....	30
2.2.2 Potential Programmatic Contributing Analyses.....	32
2.2.3 Potential Technical Contributing Analyses.....	34
2.2.4 Summary of Potential Contributing Analyses.....	39
2.3 MBSE-Optimization Integration.....	40
2.3.1 MBSE and Optimization Integration Structure.....	40
2.3.2 Variability Block Definition Diagram.....	42
2.3.3 Global Optimum Verification.....	44
2.4 Software Implementation.....	46
2.4.1 Simulation and Integration Environment.....	46

2.4.2 Architecture Modeling Tool .....	47
2.4.3 Contributing Analyses Applications.....	48
2.4.4 Optimizer .....	48
2.4.5 Setup Overview .....	49
2.5 Uncertainty and Sensitivity Analysis .....	50
2.5.1 Decision Uncertainty .....	50
2.5.2 Subjective Measurement Uncertainty.....	50
Chapter 3: Case Study 1: Satellite Communications SYSTEM OF SYSTEMS .....	53
3.1 Case Study 1 Introduction.....	53
3.2 Case Study 2 Research Setup.....	55
3.2.1 Develop a Basic Communications Satellite RA.....	55
3.2.2 Investigate Contributing Analyses and Data Sources.....	60
3.2.3 Setup of the Optimization Problem .....	61
3.2.4 MBSE and Optimization Integration.....	64
3.2.5 Contributing Objective Analyses Selection.....	66
3.2.6 Optimization Software Implementation .....	73
3.3 Case Study 1 Results.....	75
3.3.1 Simulation Output .....	75
3.3.2 Preliminary Validation .....	77
3.4 Case Study 1 Discussion .....	79
3.5 Case Study 1 Conclusion.....	81
Chapter 4: Case Study 2: Remote Sensing .....	83
4.1 Case Study 2 Introduction.....	83
4.1.1 Current Analysis of Alternatives Process.....	83
4.1.2 Opportunities for Improvement.....	83
4.1.3 Problem Statement.....	84
4.1.4 New AoA Approach Application .....	85
4.2 Case Study 2 Research Setup.....	86
4.2.1 Evaluation Method Grading .....	86
4.2.2 Overview of Scenario and Reference Architecture .....	87
4.2.3 Optimization Setup.....	91
4.2.4 Contributing Analysis Selection.....	93
4.2.5 Optimization Software Implementation .....	95
4.3 Case Study 2 Results.....	95

4.3.1 Simulation Output .....	95
4.3.2 Evaluation Methodology Comparison.....	96
4.4 Case Study 2 Discussion .....	97
4.5 Case Study 2 Conclusion.....	99
Chapter 5: Case Study 3: Mission Control SOA .....	100
5.1 Case Study 3 Introduction.....	100
5.1.1 Case Study Focus.....	100
5.1.2 Problem Statement.....	102
5.2 Case Study 3 Research Setup.....	105
5.2.1 Mission Control Segment Reference Architecture Generation .....	105
5.2.2 Optimization Setup.....	112
5.2.3 Contributing Analysis Selection.....	114
5.3 Case Study 3 Results .....	125
5.4 Case Study 3 Discussion .....	128
5.4.1 Architecture Selection .....	128
5.4.2 Satisfaction of Case Study Focus Areas .....	128
5.4.3 Utility in Assessment Flexibility and Excursions.....	130
5.4.4 Optimization and Uncertainty Integration.....	135
5.5 Case Study 3 Conclusion.....	138
Chapter 6: Summary .....	139
6.1 Synthesis of Results .....	139
6.1.1 Case Study Summaries.....	139
6.1.2 Preliminary Validation .....	140
6.2 Conclusions Derived .....	141
6.2.1 Enhancement and Comparison to Prior Methods.....	141
6.3 Recommendations for Future Work.....	143
6.4 Disclaimer .....	145
References.....	146



## LIST OF TABLES

Table 1: Example Potential Contributing Analyses.....	39
Table 2: CommSat Architecture Requirements .....	56
Table 3: Case Study 1 Objective Function Weighting Factors for Parameters of Interest.....	64
Table 4: Case Study 1 Optimum Architecture Results .....	76
Table 5: Case Study 1 Influence Factors for Objective .....	77
Table 6: Case Study 1 Influence Factors for Annual Architecture Cost.....	77
Table 7: Case Study 2 Methodology Comparison .....	96
Table 8: Combined Mission Control Segment Requirements .....	107
Table 9: Case Study 3 Objective Function Weighting and Normalization Factors for Parameters of Interest.....	113
Table 10: TechSAT MCS Development Cost Functional Breakout.....	116
Table 11: SOA MCS Development Cost Functional Breakout .....	117
Table 12: Stovepiped Cybersecurity Risk Likelihood Metrics.....	122
Table 13: SOA Cybersecurity Risk Likelihood Metrics.....	122
Table 14: Cybersecurity Risk Impacts for Stovepiped Architecture .....	123
Table 15: Cybersecurity Risk Impacts for SOA .....	123
Table 16: Stovepiped vs. SOA Objective Value Metrics .....	126
Table 17: Beta Distribution Parameters for Red Team Certification .....	132
Table 18: Stovepiped vs. SOA Objective Value Metrics for Passed Red Team Assessment ....	133

## LIST OF FIGURES

Figure 1: Example Block Definition Diagram.....	24
Figure 2: Example Internal Block Diagram.....	26
Figure 3: Example Activity Diagram.....	27
Figure 4: Example Use Case Diagram.....	28
Figure 5: Meta-Model of Notional MBSE-Optimization Integration.....	41
Figure 6: Example Variability BDD.....	43
Figure 7: ModelCenter Example Implementation .....	49
Figure 8: BDD of a Communications Satellite Domain .....	57
Figure 9 IBD Communications Satellite Operational Context .....	58
Figure 10: System-of-Systems Enterprise Diagram .....	59
Figure 11: Activity Diagram of Terminal Establish Link Request.....	60
Figure 12: Case Study 1 Meta-model of Contributing Analyses.....	65
Figure 13: Case Study 1 Variability BDD in SysML .....	66
Figure 14: Case Study 1 Parametric Diagram in SysML.....	74
Figure 15: Case Study 1 ModelCenter Implementation of Overall Objective Function .....	75
Figure 16: Case Study 1 Overall Architecture and Programmatic Optimization Process .....	81
Figure 17: IRSat Domains Composition.....	89
Figure 18: IRSat "ConductSensingOp" Activity Diagram .....	90
Figure 19: IRSat Variability BDD .....	91
Figure 20: Case Study 2 ModelCenter Simulation Setup.....	95
Figure 21: Case Study 2 Representative Current Methodology AoA Output .....	97
Figure 22: Case Study 2 Representative New Methodology AoA Output, Cost (\$M) vs. Objective .....	98
Figure 23: Case Study 3 Independent TechSAT MCS BDD.....	110
Figure 24: Case Study 3 SOA MCS BDD .....	111
Figure 25: Case Study 3 Variability BDD .....	112
Figure 26: Case Study 3 Histogram of SOA Objective Results .....	125
Figure 27: Case Study 3 Histogram of Stovepiped Architecture Objective Results .....	126
Figure 28: Case Study 3 Objective Histogram Comparison.....	127
Figure 29: Comparison of Red Team Passed and Original Architectures Objective Values .....	133
Figure 30: Objective vs. Acceleration Scaling Factor .....	137
Figure 31: Comparison of Existing to Proposed Methodology .....	142

## CHAPTER 1: INTRODUCTION

This chapter provides background information to frame the rest of the dissertation

### **1.1 Content of the Dissertation**

This dissertation presents a proposed methodology to conduct architecture selection decisions that occur early in a system acquisition in order to produce better value while also being more transparent and trusted by the stakeholders. The overall content of the dissertation is organized as follows.

Chapter 1 provides the background for the investigation starting with an overview of the problem scenario and frequent shortfalls in this vital early activity of the system acquisition process. It then presents a literature review to highlight techniques developed in various fields that may contribute to the solution space. This includes an overview of Model-Based Systems Engineering (MBSE), optimization techniques, the integration of MBSE and optimization, and uncertainty analysis within optimization. Chapter 1 closes with a proposed solution to the problem that will be evaluated in the rest of the dissertation.

Chapter 2 presents a specific implementation of the proposed methodology, including an exemplar technical execution process with associated tools. The topics covered include reference architecture generation, contributing analyses selection, MBSE integration, software implementation, and uncertainty and sensitivity analysis. This provides a baseline for the proposed methodology that will be executed through three case studies.

Chapters 3-5 present the three case studies to evaluate the proposed methodology described in Chapter 2 and include a case study background, research setup, results generated, and a discussion of those results. Chapter 3 presents a satellite communications system-of-systems (SoS)

acquisition case study to demonstrate the overall utility of the proposed methodology. Chapter 4 presents a remote sensing case study with a focus on the specific U.S. DoD Analysis of Alternatives (AoA) process to highlight how the proposed methodology directly compares with the current methodology for architecture evaluation and selection. Chapter 5 examines a mission control service-oriented architecture (SOA) with a focus on cyber security design that highlights the potential payoffs of the uncertainty and sensitivity analysis within the proposed methodology. Overall Chapters 3-5 present a thorough exercise of the proposed methodology through a range of scenarios that demonstrate its utility and benefit over the current methodology. While the case studies are focused on U.S. DoD and other large acquisition examples, the methodology explored is broadly applicable to any system design scenario where an optimized and agreed-upon architectural context is required for success.

Chapter 6 presents a summary and final contributions of the dissertation. This includes a synthesis of the results of the case studies, specific conclusions derived, and recommendations for future work.

## **1.2 Problem Overview**

In major system acquisitions an early step is an architecture alternatives evaluation and selection of the best architecture for the program to acquire. Architecture evaluations are performed to compare candidate solutions for the system acquisition on their quality and ability to address stakeholder concerns [1], such as technical performance measures and affordability. It is a critical early step that has great leverage on the overall success of the ensuing program. An exemplar of such a process is the Congressionally-mandated AoA process for Major Defense Acquisition Programs (MDAPs), which evaluates materiel solutions on operational effectiveness, suitability and life-cycle costs in order to meet capability needs [2].

Unfortunately the complexities of modern major systems can make architecture evaluations difficult. There are almost always many competing stakeholders, each with a different prioritization of objectives for the system. Different stakeholders may also use models with different semantics, leading to inconsistencies in understanding [3]. The raw technical complexity of the system can also make a comprehensive understanding of the problem space difficult for many decision makers, who are often very senior personnel with a large scope of responsibility and have to rely on trusted advisors and clear discriminators to inform their decisions. Lastly, depending on the type of evaluation, frequently the scope and objectives of the problem space change mid-evaluation as world events occur and other competing technologies become available and develop stakeholder champions.

Overall these conditions serve to create a scenario that makes it difficult to execute an architecture selection that consistently selects the best value as defined by the decision makers. While there is voluminous guidance on how to conduct an architecture evaluation in many organizations, especially in Government, how the final selection decision is determined is always up to the key decision makers. The conditions described then lead to decisions that frequently are determined by subjective measures such as which stakeholder can make the most persuasive argument during the critical decision meeting or become overly focused on one quantified measure, which is typically cost. This will then lead to decisions that don't provide best value and lack transparency for participants in the critical decision meeting, and can stymie implementation of the strategy selected.

Given the recent rapid and widespread technological advances within the fields of decision support tools, operations research, architecture modeling, and systems engineering, a better

methodology to inform architecture evaluations and selection for major system acquisition can and should be developed, a proposed approach for which is presented in this research.

### **1.3 Literature Review**

The following Literature Review was conducted in order to investigate solutions to this problem.

#### *1.3.1 MBSE Overview*

Model-Based Systems Engineering is a significant change in the fundamental way Systems Engineering (SE) is conducted in order to manage the technical baseline of a program. Traditionally, a multitude of documents are used to define critical requirements, capabilities, interfaces, and other design features of a major system. This has come to be colloquially known as “document-based SE” as documents are the authoritative materials to be carefully controlled, coordinated, and built to through configuration management processes. A significant issue with document-based SE is that as systems grow in complexity, often it is required to maintain a multitude of documents in order to describe overlapping requirements, capabilities and interfaces that must be tightly controlled and coordinated through several different organizations and processes. This significantly raises the risk that changes will not be fully captured and understood until an issue is discovered during test or operations. For instance, an update to change the telemetry data format of a satellite would have to be carefully coordinated and could involve changing system specifications for the spacecraft itself, the mission control segment terminals, the flight and mission control software, multiple interface control documents, and a range of test and contractual documentation, with each document change representing an opportunity for the update to be misunderstood, implemented incorrectly, or implementation overlooked completely.

MBSE alternatively utilizes models to control the technical baseline of the program. In its most pure form, a MBSE management implementation would use a single linked model that captures all requirements, capabilities, interfaces, and other necessary information to describe the system under development. Any coordinated changes would be implemented in the model where their affects across the architecture would be instantly captured and accurately reflected in artifacts created from the model. If documents were needed, such as to define a contractual requirement, they could be instantly and efficiently generated from the model, up to date with all changes incorporated. This ensures consistency across all descriptions and views of the system, greatly reducing risk and saving effort when compared to document-based SE [4].

Implementing a MBSE strategy is not without difficulties however. Utilizing models to control the technical baseline can be less intuitive for some participants in the acquisition process than using documents, resulting in the models generated being used as end-product descriptions rather than the core of the technical management process, thereby defeating the purpose [5]. There are also competing tools and languages. Furthermore, MBSE requires additional training and software tools to support the MBSE implementation, which is not without cost. While there are many successful implementation examples, heightened concern also surrounds MBSE interactions with non-technical disciplines.

There can be particular difficulties where MBSE must be implemented across contractual boundaries, which has traditionally relied upon copious documentation and involved supporting procurement specialists and business practices that may have difficulty integrating the models [6]. Some also attempt to pursue MBSE as a trendy method in order to make up for a poorly-implemented SE function, discovering too late that no amount of modeling software can overcome a lack of proper SE discipline. In fact common challenges with implementing models in design

activities are similar to those experienced by document-based processes, and include change management, requirements management, and user participation [7]. Given these reasons and the additional initial cost to pursue MBSE, some feel that document-based SE may be the safer option for many organizations [5].

Despite these concerns, MBSE has clearly provided major value when implemented correctly and has been gaining momentum as the technical management process of choice for leading technical development and acquisition organizations tackling complex systems. It has been broadly studied and successfully applied to a number of different disciplines and fields, including test and evaluation [8], information and embedded systems [9], and space systems [10]. The International Council of Systems Engineering (INCOSE) has committed to MBSE and has multiple working groups pursuing the development of guidance for practical MBSE implementation [11].

MBSE's effectiveness in flexibly and explicitly addressing many of the challenges associated with design problems has allowed it to efficiently integrate activities for conceptual and creative development efforts with demonstrated payoffs [12]. The impact is real for an organization's bottom line. A wide-ranging study of MBSE implementations by Sandia National Laboratory found that transitioning to a rigorous MBSE process through the lifecycle of a system development effort resulted in a "significant advantage" over document-based SE primarily from preventing defects, reducing rework and associated cost, and shortening design and acquisition schedules [13].

Specific implementations of MBSE can vary with modeling language, software tools, and architecting techniques. A widely accepted modeling language for MBSE is the Systems Modeling Language (SysML), which has widespread familiarity, applications, and software tool support



[14]. Techniques utilizing SysML have been developed for a number of complex system applications [15].

An exemplar MBSE architecting process is the Model-Based Systems Architecting Process (MBSAP), which is based on SysML [16]. The architecture can be organized through the use of operational, logical, and physical viewpoints [17]. A particularly common MBSE technique is developing a generic architecture for the problem space known as a Reference Architecture (RA), which can facilitate robust trade studies by serving as a baseline starting point for excursions that represent specific implementations of the RA. This has been shown to effectively decrease errors, development time, and cost [18].

MBSE's flexibility has been demonstrated through its wide integration with other SE management techniques. For instance, it has been combined with a Design Structure Matrix (DSM) to create a Model-Based DSM (MDSM) [19]. In particular, the flexibility gained from MBSE has been successfully applied, perhaps most critically, to the dynamic environment of early system design [20].

### *1.3.2 Optimization Techniques in Engineering*

The use of optimization techniques to aid in decision making has been around for a considerable time (for a classic overview see [21]). This field has recently expanded extensively when applied to complex engineering problems. In particular, the ability to select the “best” solution given a set of competing objectives, known as “multi-objective optimization,” is a very desirable capability because of the competing demands in engineering modern systems, such as cost, reliability and performance [22]. For a useful survey of multi-objective optimization techniques see [23].

Frequently the constraints presented in modern engineering optimizations include non-linear and non-differentiable functions. An example of such a case is the use of step functions in cost modeling to account for specific equipment package options for the system being designed. Problems that include such functions can be much harder and sometimes impossible to directly solve analytically. This can drive alternative methods to solve the problem, a popular approach for which is to utilize an evolutionary or “genetic” algorithm which leverages a machine learning feedback loop to exercise the problem space with potential solutions in an attempt to evolve the optimum solution [22].

For their flexibility and availability through numerous software tools, genetic algorithms have become a ubiquitous component in attempts to solve the extremely complex modern engineering optimization problems that have ever-increasing sophistication [24]. Their ability to handle multi-objective optimizations has been successfully applied to a wide range of engineering fields [25]. Recent research has focused on making genetic algorithms more computationally efficient through the use of parallel processors, which can greatly speed up the optimization process [26]. The latest techniques have investigated coevolutionary algorithms working cooperatively to tackle problems that have too many objectives to optimize efficiently with a single optimization algorithm [27]. Concurrent optimizations have enabled a number of creative strategies, to include varying a hierarchy of meta-models in order to solve complex optimizations in a more computationally efficient manner [28].

### *1.3.3 Applications of Optimization to System Design*

Optimization has been successfully applied to architecture evaluations for many system design scenarios [23], in most cases informing the architecture selection decision rather than determining it. While there are some exceptions, most examples focus on optimizing system

performance for a given cost or optimizing cost for a required performance. This can include very detailed cost modeling through the subsystem level, evaluation and comparison of discrete component modules, parametric relationships of technical performance, and system operational context modeling [29]. For most complex systems this is inherently an interdisciplinary endeavor, relying on component models from very different engineering or scientific fields [26]. Given the rise of computational power, the limits on what can realistically be included in an evaluation, in both breadth of options considered and depth of detail, has greatly increased. It has also enabled many alternative optimization methods to be investigated, including varying the mathematical structure of the objective function itself [30].

Genetic algorithms have been applied to system design, and in particular spacecraft design, for decades, some of the early examples of which focused on assessing component technology for incorporation into the final design [31]. Specifically, architecture evaluators felt this was a useful technique in forcing designers to break out of fixation on designs they were comfortable with. Since then, there are now numerous examples of engineering optimization occurring in just about any system design scenario, including everything from submarines [32], to launch vehicles [25], to RF sensors and information systems [33].

It's been postulated that all system architecture trade studies are fundamentally multi-objective optimizations with the essential struggle being how to represent stakeholder priorities mathematically [34]. In particular, there are frequently driving critical assumptions that can drastically affect the structure of the objective function and the priorities of the competing criteria. This most often results in many differences of opinion amongst the stakeholders about how accurately the given objective functions represent their respective desires for the system under design and what should be done to improve them. Despite these concerns, attempts to derive

mathematical objectives to aid in architecture evaluation and selection decisions are frequent. In particular, they can contribute by enabling the identification of “knee in the curve” points on Pareto frontiers (essentially local optimums between competing objectives) and emphasizing the corresponding architecture alternatives to decision makers, which in itself is a useful activity to inform further iterations of the analysis and the final selection [35].

While it is probably not acceptable to many stakeholders to leave the entire decision about an optimum architecture in the hands of a calculation, at least attempting to define a mathematical objective can be an illuminating activity [34]. Specifically, by getting stakeholders to define their relative priorities for the various decision criteria by forcing the documentation of an objective function ensures transparency, traceability, and repeatability in the decision process. In fact, other constructs have been proposed specifically to enforce traceability such as through the use of a rule-based value determination which has been established to be helpful in a varied assortment of decision support tools [36]. A mathematical objective can serve a very similar purpose, with documented changes to the objective serving as a record of the shifting priorities of the stakeholders. This provides insight into each stakeholder’s relative priorities which can help facilitate an informed discussion during the final architecture selection.

This work assumes that systems architecting is ultimately about achieving client satisfaction [37]. Interestingly enough that has traditionally resulted in the view that systems architecting is more of a qualitative “art” rather than a quantitative “science” such as systems engineering [37]. The author seeks to blend the two in this research and indeed show that by leveraging quantitative measures in architecting we can achieve better stakeholder satisfaction.

#### *1.3.4 MBSE and Optimization Integration*

The integration of optimization techniques with the comparatively newer processes of MBSE was a logical step in the maturation of system design methodologies. In fact, the combined management of system modeling with other engineering discipline models has been identified as a key part of realizing the benefits from MBSE [38]. The potential advantages posed by this integration are great, and it has been demonstrated in practice that optimization tools leveraging modeling techniques can evaluate 500 times more potential architectures than the more manual methods of a traditional architecture evaluation in the same timeframe [39]. The drawbacks of the comparatively higher learning curve and tool access have been mitigated as both MBSE and optimization techniques have demonstrated track records of utility in a variety of scenarios which lead to flexible and accessible software tool support and a growing cadre of knowledgeable practitioners.

With its popularity in MBSE, SysML has become one of the main tools to facilitate optimization integration. SysML has a metalanguage base, which makes it possible to directly integrate with a number of optimization and simulation tools [40]. There are numerous examples of this, such as a SysML integration with the space domain-focused Satellite Tool Kit [41]. Furthermore, while SysML is not an executable language itself, it can enable executable simulations through model transformations, parameter exportation, and automated code generation [42]. In fact, most mainstream SysML tools directly support simulation of behavior diagrams. Additionally, since it is a language for high level architecture modeling, SysML can be an effective integration tool between different modeling environments [43]. Despite this potential for interoperability, there are still challenges to implementation in practice [33].

In the methodology employed in this research, an architecture evaluation that leverages optimization starts with defining requirements for the system under design. Next, a trade study is developed that translates these requirements into constraints, thresholds, and mathematical relationships integrated into an overall objective function. This can include both technical parameters such as measures of system performance, and programmatic parameters such as cost and development schedule. Then, a corresponding RA is developed that encompasses the various options, which has been demonstrated in SysML [32]. It is all integrated through a simulation that links the RA with the optimization of the objective function, varying the objective and architecture until the optimum and corresponding architecture are selected. This entire process has been demonstrated through the use of Mathworks MATLAB® and Microsoft® Excel analyses linked to a SysML architecture and exercised in a Phoenix Integration ModelCenter® simulation environment through the use of Application Programming Interfaces (APIs) [33].

This methodology of integrating a SysML MBSE implementation with an optimization is actually fairly straightforward given the data elements defined by SysML. SysML utilizes a “Measure of Effectiveness” stereotype that can be an input to an objective function. Element dependencies that are defined by performance relationships can effectively be modeled through a Constraint Block, and can be very simple or extremely complex relationships. A general system block can be varied in order to represent multiple architecture configurations. Then Parametric Diagrams are used to model the constraint relationships. This flexibility can significantly aid in designing for adaptability since many modules and components can be compared and evaluated quickly. [33]

The extensibility of this type of integration between optimizations and MBSE through a simulation engine is limited only by available computational power and the available effort and

understanding of the modelers. Extremely detailed and thorough satellite constellation optimizations that include bottoms-up cost models down to the subsystem level and robust technical performance models have successfully followed this implementation [29]. A main strength of a simulation engine that incorporates API's, especially if it supports writing a tool-specific API such as ModelCenter does, means that any model defined in any tool of choice can be integrated into the overall simulation to be exercised and optimized through software calls. Alternatively, the structure of the objective can also be modified through variations of the arrangement of the Constraint blocks, allowing for other optimization strategies [30]. Certainly we are far from realizing the limits of the applications of these flexible tools and strategies.

Unquestionably the ability to integrate MBSE with optimizations has demonstrated utility in a variety of scenarios. Not only does it enable the exploration of an expanded trade space, but it also enables greater insight into how the selection of the “best” architecture occurs. In fact, it has been suggested that SysML conceptual data models be used to ensure consistency and traceability for the data in complex system architecture evaluations [44].

### *1.3.5 Uncertainty Analysis in Optimization*

#### *1.3.5.1 Tracking Uncertainty Through Modeling*

The ability to track uncertainty through an optimization is critical in ensuring an understanding of the confidence level of the final result. In particular, stakeholders will want to know if the architecture corresponding to the identified optimum solution will be likely to return value close to what was predicted at the optimum in the model (a more robust architecture), or has a greater potential to return a significantly lower value than what is predicted (a more fragile architecture). By rigorous analytical accounting for uncertainty, modelers can give increased confidence in the results.

A basic consideration concerns uncertainty in the data itself, especially in measurements of physical systems. When calculating the likelihood that a system or subsystem will meet a necessary threshold, a figure of merit known as a *k*-factor is typically used. This is usually defined by margin divided by uncertainty. A Gaussian or Normal distribution is typically assumed for the uncertainty, however that may not always be an accurate assumption. If Gaussian uncertainty cannot be assumed, then more complex measures have to be taken to estimate and bound for uncertainty. Such methods, such as utilizing tolerance intervals rather than confidence intervals, have been demonstrated to allow for the statistical analysis of all types of data even for those that do not follow a Gaussian distribution. [45]

One direct approach to account for uncertainty in an optimization is selecting uncertainty or risk as one of the criteria in the objective function itself. Techniques have been demonstrated for this such as mean-variance optimization to optimize a given return for risk, which was developed originally in the 1950s for the financial sector [46]. These techniques have been applied to a SoS architecture design optimization in order to optimize expected performance for development time risk [47]. This would require a statistical quantification of the risk of all the inputs for the objective function as well as limiting the contributing analyses to only those relationships formatted to quantify uncertainty.

Risk may not be one of the criteria desired to be optimized. In this case, as long as the risk is understood and can be quantified for all the inputs and relationships, then it can be rigorously propagated through the simulation [48]. This will allow for the determination of uncertainty bounds for the final optimized result and will give stakeholders an understanding of the likelihood the architecture will deliver on its predicted performance. However, this can be difficult in practice



because not all the inputs or relationships may be statistically understood. Furthermore, such analysis requires additional work and expertise from the modelers and others.

Another common technique to account for uncertainty is Monte Carlo analysis. This would require understanding the potential input distributions and variability of all the relationships being optimized. A typical Monte Carlo analysis application would follow four steps. First the system logic is formalized, which establishes the relationships between the parameters to be varied and the output. Next, probability distributions are assigned for each variable, which can be based from empirical historical data or known distributions. Then the probability distributions are converted to cumulative probability distributions with the cumulative probability on the ordinate to correspond with a random input. Finally, the Monte Carlo process is run in accordance with the formalized logic, with each run selecting a random number corresponding to each parameter which evaluates that parameter based on the cumulative probability distribution, ultimately resulting in an output according to the logic. A sample set of runs will then generate a distribution for the result with the validity of this distribution corresponding to the fidelity of the logic, the accuracy of the input distributions, and the number of trials in the sample. [49]

A basic implementation of a Monte Carlo analysis in an optimization would first conduct the optimization to identify the optimum set of parameters, then use those corresponding parameters with appropriate input distributions in the Monte Carlo simulation to recalculate the objective. This will give a distribution of the expected return for the originally calculated set of optimum parameters. A wider distribution, or one with many of the results, may show that the optimum solution does not often deliver on its promised value, and may warrant a re-evaluation of the objective.

There are other new methods to account for uncertainty. For instance, unscented transformations have been proposed and demonstrated for some problems as a less computationally demanding alternative to Monte Carlo to describe the effects of uncertainty within the optimization [50]. Another method is reliability-based optimization, which uses both deterministic constraints and reliability constraints in the objective function. The reliability constraints capture probabilistic failure modes and ensure they are below thresholds acceptable to the stakeholders [51]. Both these efforts demonstrate creative ways to capture uncertainty given limited knowledge about the variability in the scenario and limited computational power, which are very common concerns.

#### 1.3.5.2 Appropriateness of “Subjective” Measurements in Modeling

One of the main reservations stakeholders have with architectural modeling to make decisions is accounting for architectural aspects that are typically thought to be very subjective or nebulous to quantify, an example of which is cybersecurity [52]. This is an understandable concern as the shortfalls of human judgment in attempting to quantify uncertainty in decision making, namely the tendency to replace statistical principles with biologically-ingrained heuristics, has been robustly documented [53]. However, these factors can be better understood, and successfully compensated for through careful analysis [52], a recent example of which is highlighted in the high profile book and Hollywood movie *Moneyball* [54].

Leveraging subjective human measurements is actually a perfect application of uncertainty analysis since according to the “subjectivist” or “Bayesian” interpretation of statistics most decision makers hold (whether they realize it or not), probabilities are an attempt to quantify lack of knowledge about a possible outcome [55]. In that sense, a 90% confidence interval represents a 90% probability of containing the true value whether it was determined by a human judgment or

a physical instrument. In fact, it is precisely a result of Bayesian theory that an expert judgment should be viewed as just another measuring tool (albeit with typically a comparatively wider confidence interval) that provides a measurement with uncertainty bounds [52].

The opposing philosophical view in statistics to the “subjectivist” view is known as the “frequentist” view. It holds statistical probability can only apply to measurements that are purely random, strictly repeatable, and have an infinite number of iterations. Subjective human judgment would obviously not fall into this category, but then neither would any real world measurement no matter how precise the instrument. In this view probability is purely a mathematical abstraction. [52] It may seem hard to understand how this could be, but in the real world, there is always a chance an instrument could be mis-calibrated, misapplied or otherwise wrong. For instance, the author has personally experienced a precise technical instrument misused and holding a multi-billion-dollar aerospace system at risk because a human mistakenly applied the Celsius scale to a Fahrenheit-calibrated tool. There is no such thing as a purely objective measurement in the real world no matter how careful or sure we may think we are [52].

It is asserted then that the problem with human judgment compared to physical instruments is that human judgments are typically not calibrated very well in providing their confidence interval. Humans will tend to be overconfident with confidence interval estimates, although can be underconfident. However, they can be calibrated through training to provide accurate confidence intervals for their expert judgment measurements. This allows for the incorporation of expert judgment into quantitative techniques rather than the qualitative techniques they are typically used for. [52]

Furthermore, it has been consistently demonstrated that quantitative techniques utilizing expert judgment, even simplistic ones, consistently outperform qualitative expert judgment in

predicting results. [56] [57] The main matter is describing the information in a way that is quantifiable. While some try to provide a counter argument that there are just some things that appear too nebulous to quantify, that is never actually the case. For instance, take the situational awareness of a military user in an operations center, which may appear difficult to quantify. However, there are methods to quantify the ability to share information, and the quality of that information across a network that could serve as an appropriate model. For instance, it is possible to quantify the number of networked participants who have a common relevant operating picture (CROP) of the battlespace. [58] While it takes some thought, quantifiable metrics can be derived for all real world scenarios. [52]

In this manner, a rigorous uncertainty approach will also help gain stakeholder acceptance to architectural trades with a great deal of subjective parameters [52].

#### **1.4 Proposed Solution**

After synthesizing the results of the literature review, several potential techniques emerge to address the problem of improving the quality of architecture evaluations and selections in order to provide increased value to stakeholders while enabling better transparency into the final decision. In particular, the cited work strongly suggests that an integrated application of MBSE and optimization will lend structure and improve stakeholder feedback to enable selecting the alternative that delivers best value during the evaluation.

Given MBSE's demonstrated utility in enabling effective communication of an architecture description through its lifecycle, this research starts by applying it in a similar role at the beginning of the lifecycle early in the architecture evaluation and selection. The first step is to define a reference architecture (RA) for the solution space, including initial requirements, capabilities, and other necessary parameters. Then proposed excursions are defined including their impact on the

RA. The RA and excursions effectively communicate the boundaries of the solution space to all stakeholders.

In parallel with the MBSE effort, an optimization set up is defined. A principal component of this is to force decision makers with stakeholder input to define quantitative evaluation criteria with corresponding weightings. This provides an objective function for the optimization. While this is a significant departure from current practice given that qualitative criteria are currently frequently used in evaluations, enforcing quantitative criteria so that all relevant criteria are appropriately treated is critical. A very common occurrence in architecture evaluations now is that quantitative criteria such as estimated cost tend to overshadow qualitative criteria.

Given the criticality of the objective definition step, a robust and authoritative process must be created to execute it. This is closely related to, and perhaps simultaneous with, the requirements generation process. It is understood that stakeholders will not agree to be beholden to an analysis without first seeing the results of the analysis, so this is the initial starting point for the discussion of the objective rather than the final solution. Opportunities to iterate the objective function with the decision makers come later.

Next, the MBSE set up is integrated with the optimization. This starts with calculating the effect of the excursions of the RA for each of the objective criteria. This leads to the creation and indexing of a number of contributing analyses, each one defining a necessary step in calculating the objective criterion. Each of these contributing analyses is defined in its own discrete software implementation. A flexible simulation tool provides APIs to integrate all of the contributing analyses into one simulation scenario through the framework established in the RA with excursions in defining the objective.

The scenario is then integrated with an optimizer that exercises the excursions of the RA, calculating the corresponding contributing analyses and objective. It is likely that this optimization function will have to support non-differentiable constraints, leading to the selection of a flexible optimization tool such as a genetic algorithm. An optimum solution set of inputs variables is identified along with the architecture that corresponds to those variables. Additionally all the results of the optimization are captured to identify Pareto frontiers among the criteria.

In a real-world program, the optimum solution, corresponding architecture, and other results would be presented to decision makers during a decision meeting. If the results are not accepted, the decision makers will be forced to adjust their objective criteria and weightings, which can be informed by the Pareto frontiers identified. Any adjustment to the objective criteria and weightings are carefully documented to ensure transparency. The optimization can be iterated as often as necessary and presented to stakeholders, with any changes documented, until the results are accepted.

Once the results of the optimization are selected, the architecture identified as corresponding to the accepted solution is established as the baseline architecture for the system under design. Since this was already built in an MBSE tool, this step will just involve an adjustment to the RA to reflect the specific implementation of the excursion selection. This MBSE implementation is then incorporated into the technical baseline management process. This ensures that the architecture selected by the stakeholders is the architecture that the system designers will start building to.

## CHAPTER 2: OVERVIEW OF APPROACH

This chapter provides a detailed overview of the approach taken in the subsequent case studies for this research. Following the description of the proposed solution in section 1.4, there are five components to this approach. These include Reference Architecture Generation, Contributing Analyses Selection, MBSE Integration, Software Implementation, and Uncertainty and Sensitivity Analysis. The proposed process will be validated by executing and expanding on these components through the case studies and leveraging expert feedback, real-world comparison, and direct analytical evaluation of the merit of the solution.

### **2.1 Reference Architecture Generation**

#### *2.1.1 Reference Architecture Overview*

The first step in developing an MBSE-enabled implementation of architecture evaluation and selection is creating a RA. The RA serves as the baseline to-be architecture for the system under design. It is an abstract construct that outlines the logical and functional behavior of a class of systems. When physical detail is added to the RA, it becomes instantiated as a physical architecture for a specific system implementation. A structured process should be followed for the initial creation and instantiation process of the architecture to ensure all appropriate information is layered into the architecture while conforming to all appropriate policies and mandates. The structured process selected to perform this function for this research is MBSAP, which emphasizes an object-oriented approach for architecting in order to best implement MBSE. [16]

Following MBSAP, there are a number of activities that take place during RA development. It starts by defining the abstract behavior, structure, and other defining features of the problem space for the system under design. Next, a requirements template is built to capture

the requirements for the system that need to be addressed. Then, quality attributes are collected which define how to measure value for the architecture. The RA is then modeled in an established modeling methodology, with a preference for one that supports an objected-oriented approach. It is critical to ensure that any lessons from experience such as best practices are incorporated. The last step of RA generation under MBSAP is to validate the RA with customers, subject matter experts (SME's) and other stakeholders. [16]

### *2.1.2 SysML Introduction*

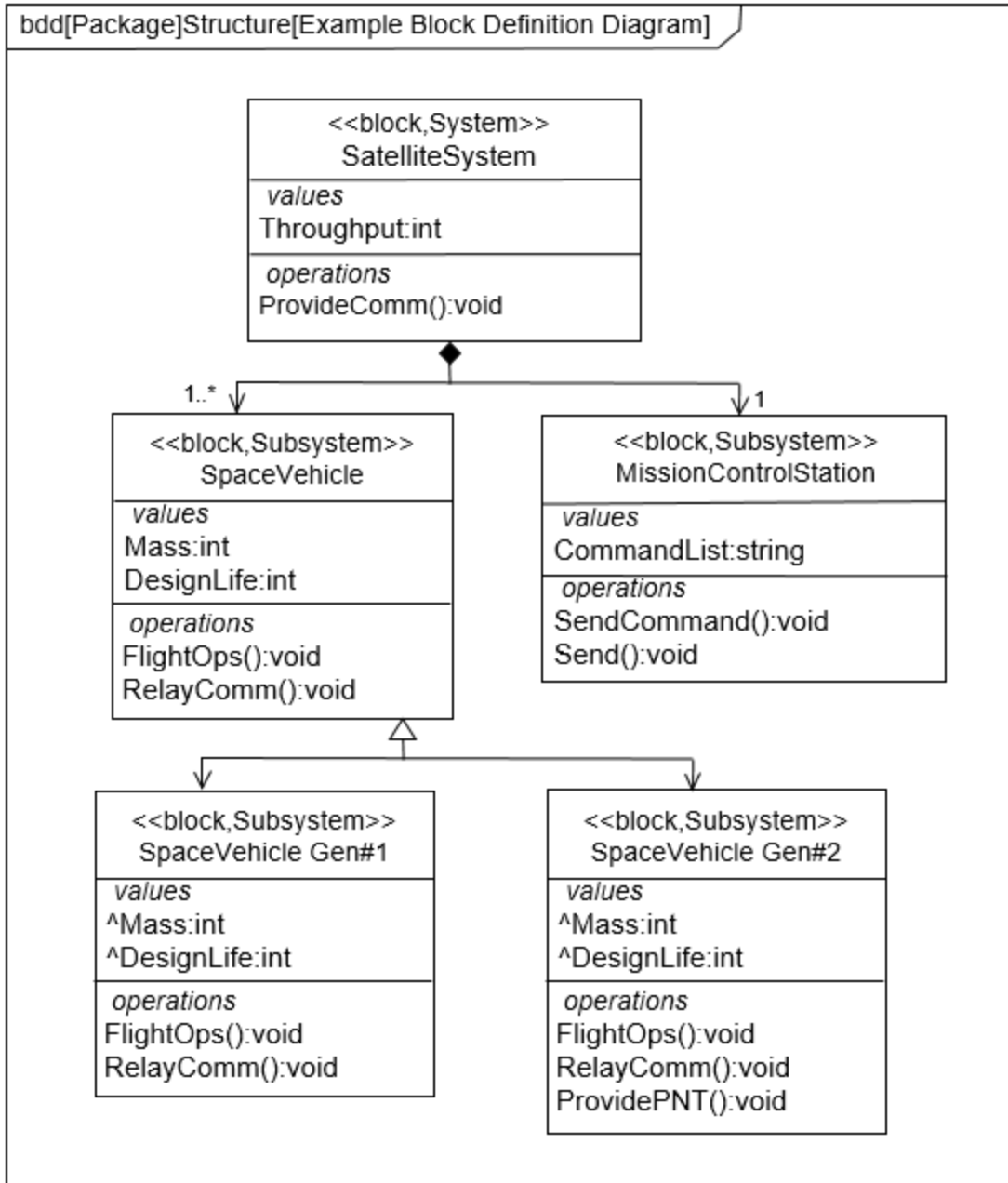
Due to its widespread use and software tool support, SysML is the architecture language selected for this research. SysML is an object-oriented modeling language that is a profile of the Unified Modeling Language (UML) developed to specifically focus on system design. It is an evolving language that is also an international standard [59]. SysML was specifically designed to support a MBSE approach in the activities of design, specification, analysis, and verification and can include hardware, software, personnel, procedures, facilities, and data. It is used to describe aspects of a system such as structure, behavior, requirements, and parametric relationships [15]. For a more complete description of the diagrams available and the SysML language in general, see references [14] [15], but a brief description of key concepts follows.

SysML utilizes nine different types of diagrams to convey information about the system, with each diagram emphasizing a different aspect of the system. However, in a MBSE approach each diagram is referencing information contained in the same underlying linked model which enforces consistency across any of these views of the system. This blend of flexibility and rigorous consistency ensures the SysML model has maximum applicability to the variety of activities associated with system design, development, and sustainment, which reduces cost and errors. The



diagrams defined by SysML include a requirements diagram, two structure diagrams, four behavior diagrams, a parametric diagram, and a package diagram.

A key concept of SysML is a block, which is a general purpose construct that may represent a component or a system. A block can contain features that represent its functions, properties, interfaces, and states. Relationships between blocks can include composite relationships, and a generalization/specialization relationship. A block definition diagram (BDD) is used to describe blocks and their relationships. [15] An example BDD is shown in Figure 1.

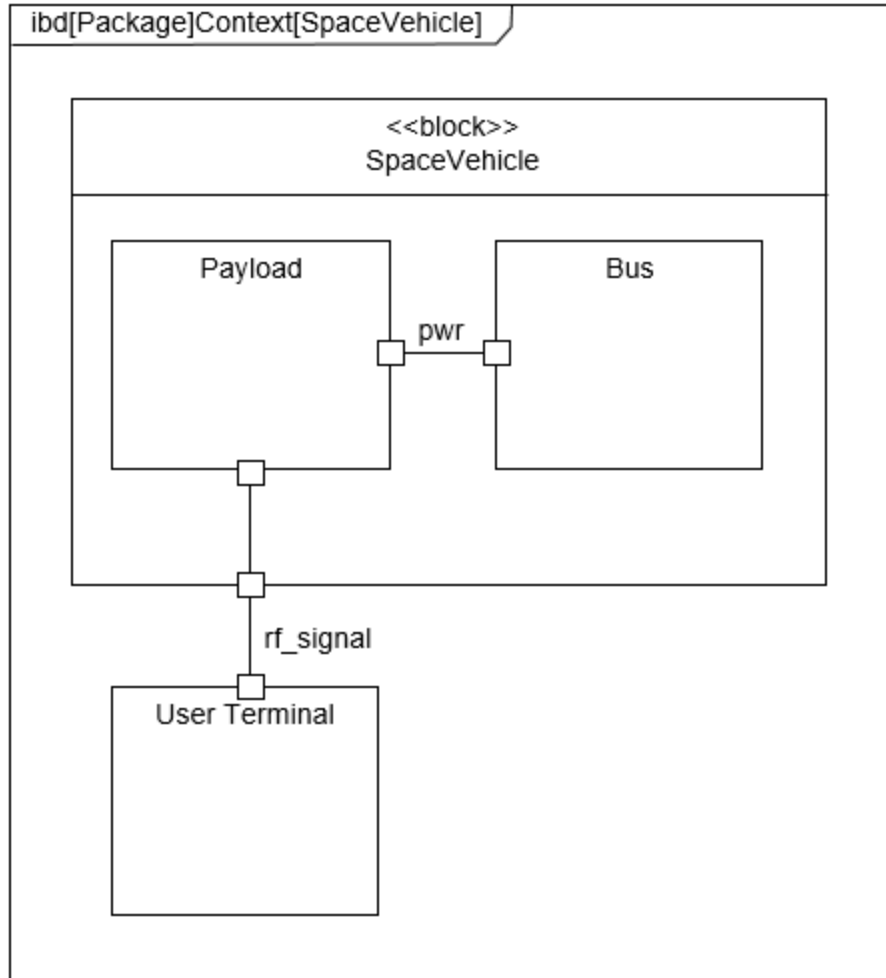


**Figure 1: Example Block Definition Diagram**

Figure 1 shows several constructs of SysML. Starting at the top is a block representing an overall satellite system. It is identified by the stereotype <<System>> and has the descriptors of values describing the system and operations identifying what it performs. Immediately below it are blocks identifying the subsystems Spacecraft and Mission Control Station which are identified as Part Properties or Parts of the overall satellite system by the composite relationship shown by a

solid diamond. There are one or more space vehicles identified by the “1..\*” multiplicity and exactly 1 Mission Control System in the overall Satellite System. Below the Space Vehicle block are two specialized blocks identified by the empty triangle showing Generalization/ Inheritance which specialize the Space Vehicle block to create two generations of Space Vehicles. These “specific” versions of the Space Vehicle have some properties inherited from the parent block and additional properties for that specific generation.

Blocks can be further broken down as interconnected elements termed parts with interaction points between blocks and parts identified as ports. A construct known as a connector connects parts. These elements are shown in an internal block diagram (IBD), an example of which is shown in Figure 2.

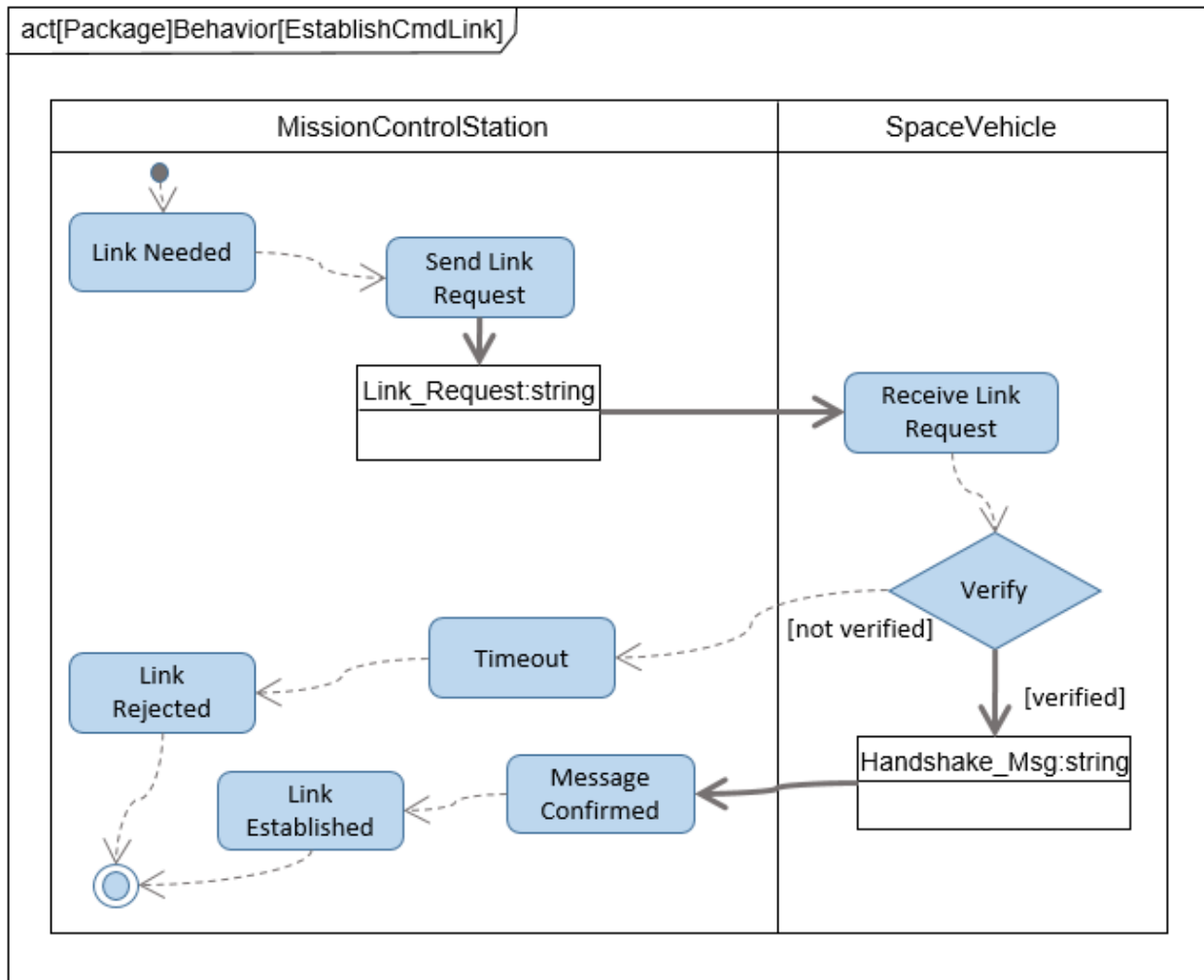


**Figure 2: Example Internal Block Diagram**

Figure 2 demonstrates several of the common elements in a SysML IBD. A “SpaceVehicle” block contains the “Bus” and “Payload” parts with the “Payload” interacting with an external “User Terminal” part. Ports are shown as small squares on the boundaries of parts or blocks. Solid lines are connectors which represents flows of matter, energy, and information such as electrical power and data carried by radio frequency energy. The IBD is very useful in showing interactions in the structure of the system.

Another important set of SysML diagrams are used to represent behaviors. In particular, activity diagrams can be used to model control flow, information object flow, input and output. Activities transform inputs into outputs through actions which are carried out in a controlled

sequence. Actions can be allocated to components which can be shown through the use of activity partitions or swim lanes in the activity diagram. [15] An example activity diagram is shown in Figure 3.

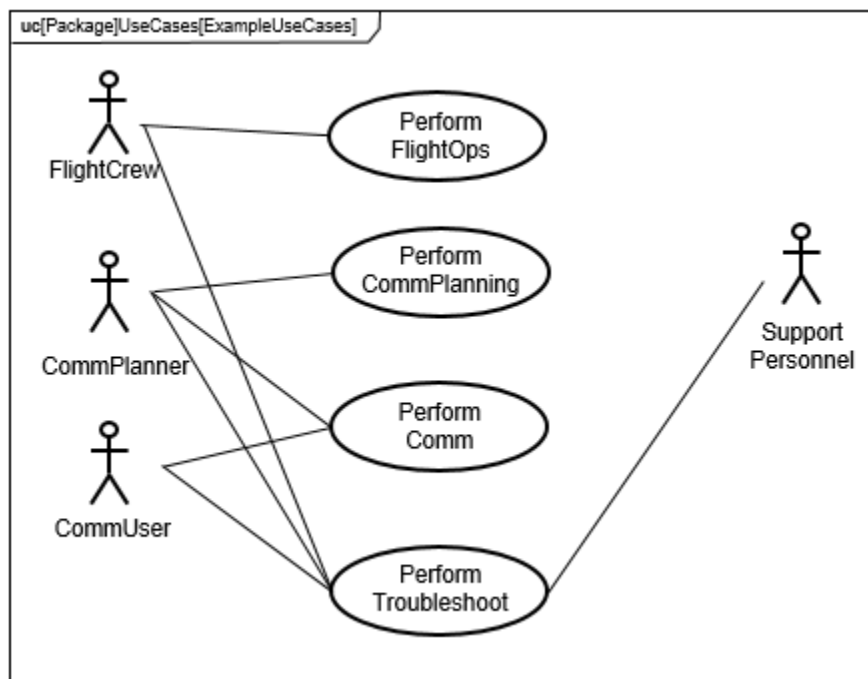


**Figure 3: Example Activity Diagram**

This example activity diagram shows the behavior interactions between the “MissionControlStation” and “SpaceVehicle” subsystem in order to carry out the “EstablishCmdLink” activity. Various actions are allocated between the two subsystems in the order dictated by the control flow. The control flow starts in the upper left with the initial node and ends in the lower left with the activity final node. The data items “Link\_Request” and

“Handshake\_Msg” are created and consumed during the course of the behavior. A diamond represents a decision gate which provides for two alternative paths for the activity to follow depending on whether or not the criteria are satisfied.

Another common diagram used to substantiate behavior modeling is the Use Case Diagram. The use case diagram is typically applied to define the overall goals of the system such as mission objectives. The goals are represented as use cases, which can be associated with the subject system and external actors such as human personnel. The use cases can then be further expanded through other behavior diagrams. [15] An example use case diagram for the example satellite system is shown in Figure 4.



**Figure 4: Example Use Case Diagram**

Figure 4 shows several of the use cases represented as ovals down the center of the diagram. These use cases reflect the top level goals of the example satellite system and each can be further expanded by an activity diagram or other behavior diagrams. They are tied to external actors represented by the stick figures. These external actors could be human personnel or an external

system the satellite system interacts with. For instance, the “CommUser” actor would include the user terminal system that communicates with the satellite, which may also include the human personnel that operate that terminal.

### *2.2.3 Reference Architecture Organization*

SysML diagrams are created to support the RA generation and organized in order to provide Operational and Logical/Functional Viewpoints [17]. A Physical Viewpoint is not created until the RA is instantiated as a specific architecture. The various viewpoints will be further broken down into various perspectives, such as the structural perspective, the behavioral perspective, the data perspective, and the services perspective, with each perspective highlighting a different aspect of the architecture.

In the Operational Viewpoint, the structural perspective will typically have generalized domains such as Planning, Information Management, and Communications Management. Common internal and external interaction points will be modeled as Ports or Interfaces on the blocks that model domains. The corresponding behavioral perspective will contain behavior modeling diagrams to identify use cases and generic user roles. The scenarios representing the flow of activities in a Use Case are modeled in Activity Diagrams. Ideally the generic operational sequence known as a Mission Thread will also be modeled in an activity diagram. The data perspective will include a conceptual data model (CDM) describing the relevant data and a services perspective, if necessary, will describe any functions that can be called as services. [16]

In the Logical/Functional Viewpoint, the structural perspective will contain any design patterns (generalized, reusable entity descriptions) for systems contained in the RA. The behavioral perspective will contain sequence and state machine diagrams describing behavior of blocks that correspond to the design patterns. It can also contain more specific timing information

for the Mission Threads. The data perspective will contain a logical data model and the services perspective will include a Services Catalog that further describes services and their specific allocation to blocks. [16]

## **2.2 Contributing Analyses Selection**

### *2.2.1 Criteria for Contributing Analyses*

Once the RA is understood and the trade space defined, the next step of the proposed modified architecture evaluation process is to select contributing analyses that can define and quantify objectives in the optimization. This selection is very problem dependent and is informed by discussion with the stakeholders and the overall requirements for the system under design. Typically top level operational requirements will be provided from users through a carefully vetted process; for instance in the U.S. DoD this will typically come from the Joint Capabilities Integration and Development System (JCIDS) which validates operational military requirements through the Vice Chairman of the Joint Chiefs of Staff [60]. In addition to meeting stakeholder goals, however, these contributing analyses must be able to integrate into the overall optimization schema in order to be acceptable.

A main factor in whether or not a contributing analysis is suitable for this process is whether it can be quantitatively measured and modeled as a metric. This ensures that it is compatible with an optimization type of methodology. Given the robust development of genetic algorithms and other flexible tools to handle non-differentiable problem spaces, the contributing analysis does not have to fit any particular form as long as it produces a quantitative metric. Even fairly robust cost analyses with complicated step functions have been demonstrated to work with an optimization tool [29].



A second factor is whether or not uncertainty data can be captured or calculated for the contributing analyses. This can be time consuming to perform, and uncertainty quantification is often only done on major projects [61], however it is critical to this process. A relatively common example is cost estimating analyses, which will typically have a predicted cost parameterized for a level of confidence. While the proposed methodology could be run without uncertainty information, the author feels strongly that being able to quantify uncertainty in the optimized solution is critical to achieving stakeholder confidence in the final result. That can only be achieved if uncertainty in the input parameters and all the contributing analyses can be quantified. Typically this means that the model used is based on and validated through large sample sizes of historical data, however techniques have been developed to generate defensible, quantified metrics with uncertainty bounds from data that comes studies of small sample sizes or subjective expert judgment [52]. While this may result in large uncertainty distributions, it is still preferable to relying on qualitative assessments.

It should be noted that not all contributing analyses directly convert input parameters into objectives in the optimization. Sometimes intermediate contributing analyses are required to calculate intermediate parameters that then feed into a later set of contributing analyses to generate the objectives to be optimized. These intermediate contributing analyses have the same requirements for quantification and uncertainty in order to be utilized in this proposed methodology.

An example of an intermediate contributing analysis could be a satellite architecture optimization that includes schedule and cost considerations. Frequently programmatic models such as these include inputs for the satellite mass, which is typically itself an output of satellite performance models [62]. So while satellite mass is not usually an objective in of itself in the

optimization, it is a necessary intermediate contributing analysis for many satellite architecture optimizations.

### *2.2.2 Potential Programmatic Contributing Analyses*

Contributing analyses based on validated historical data are attractive in this methodology because large sample sizes with many programmatic metrics naturally integrate well into this construct rather than more specific technical performance metrics. This is due to the fact that many programmatic measures are mandated by oversight authorities to be collected on many large acquisition programs. Numerous types of programmatic metrics exist and make good candidate contributing analyses dependent on the preferences of the stakeholders. These include metrics on program execution, changing requirements, and organizational relationships.

A specific source of potentially useful programmatic measures to use as contributing analyses are related to Earned Value Management (EVM), which is intended to provide leadership insight into program execution. EVM is a mandated management system on all U.S. DoD major acquisitions programs that provides reportable cost and schedule information comparing actual program execution performance to the predicted programmatic baseline [63]. Available EVM metrics include total budget, scheduled and actual expenditures, and schedule and cost variances from the approved schedule and cost baselines. With the mandated nature of EVM, these metrics are available on nearly all major defense systems, leading to large sample sizes which provides a suitable base to calculate uncertainty.

EVM derived metrics are attractive precisely because this cost and associated uncertainty and risk are calculated through a defined, repeatable process within the normal U.S. DoD acquisition processes [64]. These uncertainty analyses are typically used to inform major acquisition decisions and as such are conducted with extreme rigor. Furthermore, there are

continually efforts underway to validate cost estimation methodologies against historical data in an effort to continuously improve them [65]. Given these models already have validation and uncertainty quantification performed on them, they are ideal candidates for contributing analyses in this proposed architecture evaluation technique.

Another source of programmatic metrics are configuration changes within a Government system, especially when that system is part of a System-of-Systems (SoS). These changes must be tightly controlled, coordinated and documented, typically through a robust Configuration Control Board (CCB) process [66]. Whenever there is a commitment of funds to modify a technical baseline, there is or should be some sort of controlled CCB approval process to ensure the change does not have unforeseen ramifications across segment or system boundaries. This is a necessary component of a rigorous SE implementation because the SE model is critical in identifying such consequences of a proposed change.

While every SoS is different, in the author's experience there can be 100 or more CCB change packages a year in a significant SoS as the capabilities and requirements of the constituent systems evolve. The documentation associated with these approvals represents a wealth of information to include affected organizations, types of modifications, programs involved, contract types, and funding impacts. These documents could potentially be mined to construct suitable models for contributing analyses in the new proposed architecture evaluation methodology. These models could focus on system adaptability and potentially could have large enough sample sizes to quantify uncertainty measures.

A few additional sources of programmatic measures are available. In major acquisition programs, especially Government programs, there is often rigorous oversight requiring the generation of copious documentation, typically proportional to the size of the budget of the

program. Some examples come from the defense sector. A US Air Force program of sufficient size will have to submit a Monthly Activity Report (MAR) to its Service Acquisition Executive [67], a quarterly Defense Acquisition Executive Summary (DAES) to the Office of the Secretary of Defense (required for all U.S. DoD Major Defense Acquisition Programs (MDAPs) and Major Automated Information Systems (MAISs)) [63], and if it's an MDAP Acquisition Category (ACAT) 1 program (those with the largest budgets or acquiring the systems deemed most critical to national defense), it must submit an annual, comprehensive Selected Acquisition Report (SAR) to Congress [68]. These documents are in addition to many other management tools U.S. Air Force program managers are required to utilize to report program progress to various system stakeholders and higher oversight authorities.

In the author's experience, the MAR, DAES, SAR and other program status documentation are typically substantial products requiring significant work across the U.S. DoD. This is a benefit to this research though because many, including the MAR, DAES and SAR, are also usually readily accessible and can provide a wealth of information beyond just the EVM metrics of cost, schedule, and associated deviations. This can include program manager's and program executive officer's assessments and ratings, contractual information including type and incentive structure, risk posture, system regulatory and statutory compliance status, and interoperability status with other systems. Given the standardized reporting requirements for some of these products and associated sample size across the U.S. DoD, they make excellent sources of programmatic information to derive contributing analyses under this proposed methodology.

### *2.2.3 Potential Technical Contributing Analyses*

Identifying contributing analyses for this proposed architecture evaluation methodology that deal with technical metrics is often more challenging than those associated with programmatic

contributing analyses due to smaller sample sizes and the associated reduced understanding of uncertainty, increased specificity of performance requirements, and the sometimes nebulous traceability of an operational performance requirement to a quantifiable metric. Despite these challenges, being able to identify suitable contributing analyses that capture system technical performance requirements is essential to the success of this proposed methodology. Technical performance captures what it is that the system must do in order to contribute to an operational mission, so naturally these are critical system metrics for stakeholders.

The difficulties in determining whether a new system under development will meet operational performance goals has been directly addressed by the U.S. DoD. It is hard to predict how such a new development system will perform in an operational environment; however it is necessary to have some measure to track system development progress to ensure that the system will meet its required end state performance goals. To this end, the U.S. DoD has established a construct known as Key Performance Parameters (KPPs) to ensure progress can be measured. “KPPs are those system attributes considered most critical or essential for effective military capability.” [69] Given the criticality of KPPs, both in terms of operational utility for the system under design and financial implications for the acquisition authority, they are determined and approved through the rigorous and formal JCIDS process. [69] Additionally, private-sector product developments can readily employ measures equivalent to KPPs to measure progress against performance goals.

KPPs must be measurable and as such are quantifiable metrics [69]. KPPs for the final system will range from the minimum acceptable, known as the threshold, to a maximum above which no further benefit will be realized, known as the objective. Performance against KPPs is tracked through system development and with increasingly difficult gates until the prototype or

developmental system achieves the threshold requirement. Frequently predictive KPP models are used to assess how well a prototype is projected to mature towards achieving its final KPP thresholds. Many of these models are analytical relationships based on known physical performance calculations, but they can also be parametric models utilizing historic system performance from similar systems. In both of these cases, capturing and accounting for uncertainty is critical given what is at stake.

Some KPPs are not defined by a range but rather are either achieved or not achieved. For instance there is a set of Boolean criteria known as the “Net-Ready KPP” that ensure a system meets information exchange needs to enable operational effectiveness. This is designed to facilitate a warfighting end goal of a “cost effective, seamlessly integrated environment” which leads to requirements for interoperability [70]. Despite focusing on system emergent behavior with other systems rather than strictly the warfighting performance of the system itself, the Net-Ready KPP is treated in the same manner and overseen with the same rigor as any other KPP with the same potential consequences if it isn’t met [71]. However, due to its nature it is typically treated as a compliance item rather than a variable with a potential tradespace range like other KPPs that have a threshold and objective. However, system performance in complying with these Net-Ready KPP criteria is definitely measurable and can be tied to a metric, even if that happens to result in a binary metric quantified as compliant or not. In such a case, threshold equals objective for measuring that KPP’s performance. There are more examples of compliance-only KPPs, but the Net Ready KPP is one of the most ubiquitous.

If a system is not on track to meet at least the threshold value for a KPP for its production units, either the acquisition must be cancelled, or the program must be rebaselined to support further development activity (typically with additional funding and/or a schedule extension), or

the system requirements must be rescoped with approval from the operational command that would receive the final system. This choice may even require Congressional approval depending on the chosen solution to the deficiency. [69]

KPPs aren't perfect surrogates for operational performance and there are difficulties associated with determining KPPs that adequately cover the breadth of operational situations a new system is being acquired in order to handle. This is because it is hard to distill everything that may determine successful mission performance to a few specific metrics. Despite these difficulties, the U.S. DoD still must use these critical metrics in determining and tracking system development performance and has established effectiveness measures to ensure the system under design will meet its operational goals before it is fielded.

To prove that a new U.S. DoD system meets all operational requirements it undergoes what is known as Operational Test & Evaluation (OT&E). This process takes place typically at the end of a development period and validates that the system can perform its assigned mission in its operational environment [63]. There have been cases where a system has met all KPPs and yet failed its OT&E [72]. In these cases either there were not a sufficient number of KPPs or the KPPs selected did not adequately capture realistic operational goals.

A real world example of inadequate KPPs comes from the U.S. Navy P-8A Poseidon maritime patrol aircraft, which is meant to conduct Anti-Submarine Warfare (ASW) and other missions. The P-8A's KPPs that had initially been approved through the JCIDS process were that it only be able to fly a certain range, carry a certain number of sonobuoys (an expendable, sonar-capable buoy), and be able to communicate with certain radios. During OT&E, it was identified that even though it was meeting these KPPs, those performance requirements alone did not actually enable the P-8A to perform its mission in finding and attacking adversary submarines. Based upon

these tests, it was recommended that the P-8A program's KPPs be modified and additional development work be required to ensure the production P-8A aircraft will be able to achieve successful mission performance. This recommended change was then sent through the JCIDS process oversight authority for validation and implementation. [72]

These situations of deficient KPPs are handled on a case-by-case basis. In the author's experience sometimes the system is still accepted into operations (often with the noted operational deficiency concealed by classification), potentially with additional development work scheduled to mitigate the finding. If the situation is critical enough, the system can be held in development, with the KPPs to be updated and revalidated through the JCIDS process to ensure the finding is fixed before the system goes into production. The important take away to note is that the U.S. DoD has an established and accepted process for tying system operational performance to technical performance metrics that starts from system requirements generation through final operational acceptance.

For the purposes of this research this is significant because it shows that even something as ill-defined as warfighting ability can be assessed and tracked using as a specific set of quantifiable metrics in order to hold programs accountable for achieving specific system capabilities. The fact that such a large organization, covering such an wide range of systems and missions, and with such a large budget as the U.S. DoD, utilizes this method highlights that quantification of performance is not only flexible, but it is critical to managing major acquisitions affordably. As such, it offers an explicit counterpoint to parties that say it is too hard to quantify operational performance requirements. In fact it suggests that such quantification is indeed appropriate to use for technical performance requirements early in the system life cycle, in the initial requirements generation stage, before KPPs are defined. Quantification can be used all the



way back at the start of the acquisition cycle in the initial architecture selection and evaluation. This research explores such an approach with the goal of demonstrating contributing analysis in support of architecture optimization.

#### 2.2.4 Summary of Potential Contributing Analyses

For the setup of the new proposed architecture evaluation and selection methodology it is necessary to have supporting models to calculate objectives for the optimization function. These are identified as contributing analyses. In order to be successfully integrated into this proposed methodology these contributing analyses as a whole must adequately capture the desires of the stakeholders, and all of them must be quantifiable with uncertainty impacts understood. There may be intermediate contributing analyses that don't directly calculate an objective for the optimization but are a necessary intermediate step; the same requirements for quantification and uncertainty understanding also apply to these.

Contributing analyses can come from either programmatic data, such as cost and schedule models, or technical data, such as physics-based performance models. A selection of potential contributing analyses is shown in Table 1.

**Table 1: Example Potential Contributing Analyses**

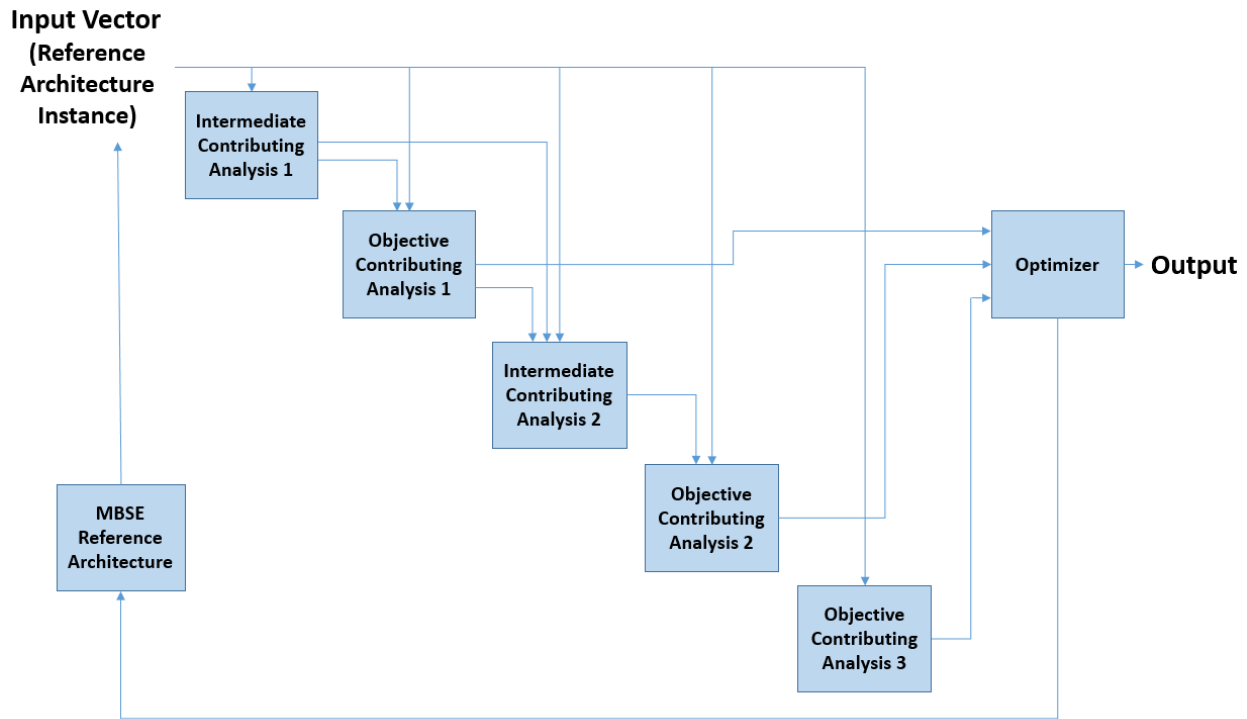
Model	Type	Source
Programmatic Models		
Predictive Cost	Parametric Relationship	Historical Program Performance Data
Predictive Schedule	Parametric Relationship	Historical Program Performance Data
Configuration Stability	Parametric Relationship	CCB Package Data
Requirement Stability	Parametric Relationship	CCB Package Data
Risk Tolerance	Analysis	Programmatic Reporting Documentation

Technical Models		
Regulatory Compliance	Analysis-Boolean	System Test, Analysis Data
Technical Performance (ex. Aircraft Speed)	Physics based Simulation	Test Data, Physical Relationships
Modularity	Analysis	System Architecture Documentation
Service Resilience (ex. Path Diversity)	Selected Analytical Relationships	Test Data, Documentation, Physical Relationships
System Supportability	Analysis	System Documentation, Test Data, Analysis Data
Cybersecurity	Analysis	Risk Management Framework Assessment

**2.3 MBSE-Optimization Integration**

*2.3.1 MBSE and Optimization Integration Structure*

Once the contributing analyses are defined, the next step is to integrate MBSE with the optimization scheme including the contributing analyses. A top level meta-model for such a notional structure is shown in Figure 5. This structure would be implemented in a simulation environment to enable the linkage of the reference architecture, contributing analyses, and the optimizer. This approach to integrate MBSE and optimization has been demonstrated and applied to system design [33].



**Figure 5: Meta-Model of Notional MBSE-Optimization Integration**

In this meta-model, a RA is the starting point. An initial instantiation of the reference architecture is generated, which generates a number of corresponding architecture parameters. These parameters then serve as inputs into a series of contributing analyses. The contributing analyses can be either an objective contributing analysis (the output of which is an objective to be optimized) or an intermediate analysis (not generating an output directly to be optimized but necessary for an input to an objective contributing analysis). These contributing analyses can be arranged in whatever sequence is necessary to ensure a logical succession of parameter calculation and generation without recursion in order to generate all the necessary objective parameters. These objective parameters are then ingested into the optimization function to calculate an overall architecture value for that particular instantiated architecture.

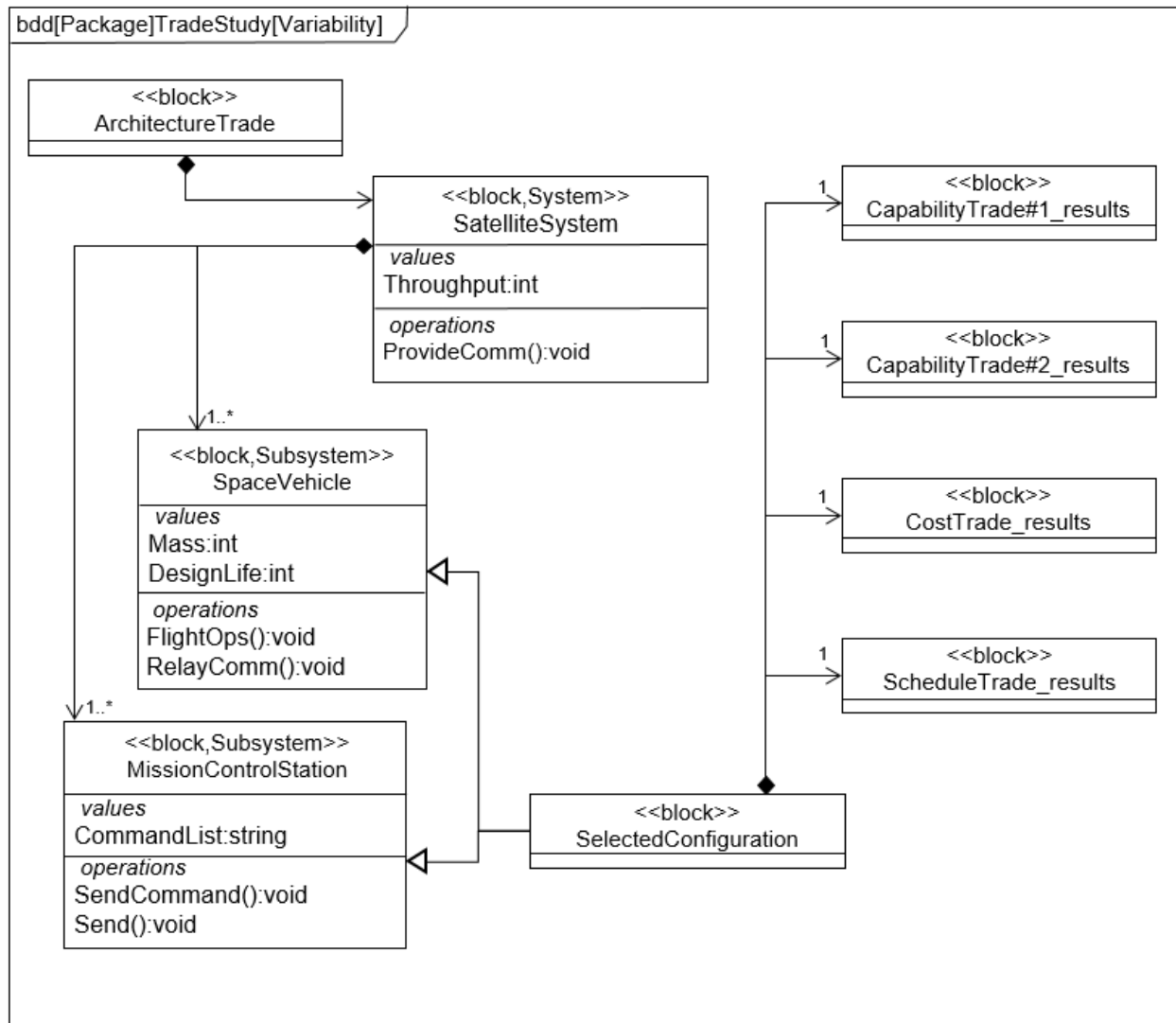
Next, the optimizer feeds back an excursion or refinement of the initial reference architecture, creating a new instantiated architecture. This generates a new corresponding set of

architecture parameters which serve as a new input vector to the contributing analyses. This then feeds a new set of objective parameters into the optimizer, calculating a new architecture value for the new particular instantiated architecture. Based on how this compares to the first instantiated architecture's calculated value and the optimizer logic, a new perturbation of the architecture instance is selected creating yet a new instantiated architecture. The cycle continues with the optimizer generating architecture variants to eventually converge on an optimum.

In the simulation environment, once the perturbations have suitably stabilized (how much convergence is "suitable" will depend on the research problem and computational resources at hand), the simulation is ended and the optimum is identified. The corresponding specific architecture that was used to generate the optimum is also identified as the ideal specific architecture [33]. That specific MBSE implementation would then be used to describe the selected architecture to stakeholders.

### *2.3.2 Variability Block Definition Diagram*

A useful construct to facilitate the integration of MBSE, in particular a SysML implementation of MBSE, with an optimizer is known as a Variability BDD. A Variability BDD facilitates modeling multiple specific architecture implementations as variants from a general system block. This is used to show traceability in the parameters to be optimized to their impact on the architecture being optimized [33]. An example Variability BDD that describes a potential architecture selection trade space using the example satellite architecture described in Figure 1 as the starting point is shown in Figure 6.



**Figure 6: Example Variability BDD**

In this example Variability BDD the architecture is shown from Figure 1 including the top level system block “SatelliteSystem” and its constituent parts, “SpaceVehicle” and “MissionControlStation”. The notional trades selected for this diagram are blocks shown down the right side of the diagram and encompass both programmatic and technical trades. The results of these trades are themselves constituent parts of the selected configuration. The selected configuration then specializes the architecture description in order to represent the configuration that was selected through the optimization of the trades.

This Variability BDD setup will work in an integrated fashion with the meta-model shown in Figure 2-5. Each of the trades shown in Figure 2-6 represents something about the architecture that can be varied and effectively serves as an input to be optimized. The simulation takes these inputs to the architecture, defines the specific architecture configuration and then calculates the various parameters necessary through exercising the contributing analyses, with the results fed into the optimizer. The optimizer then calculates an overall value for the architecture, decides which input parameters should be varied, and makes the corresponding changes, using the results of the trades shown in Figure 2-6. These trades then affect the rest of the architecture in the prescribed fashion, allowing for generation of the specific configuration of the instantiated architecture and the updated calculation of the contributing analyses for the new run. This cycle repeats until the simulation achieves completion criterion (scenario dependent, but typically completion is achieved when the in the calculated objective achieves a certain stability threshold, such as all new changes are within 1% of the total objective).

### *2.3.3 Global Optimum Verification*

It is necessary to establish confidence that the optimum identified is a global optimum and not a local optimum. Several methods can be used to do so. The most straightforward is to perform multiple additional runs of the simulation for confirmation, starting with a different initial instantiated architecture and corresponding input vector. These subsequent runs will be executed until it is clear that they are converging on the same optimum. If that isn't the case then the simulation is run until a new optimum is identified. Additional runs are conducted until suitable confidence is gained that the final result represents the global optimum. This has been demonstrated to be an acceptable method of confirmation in similar optimizations with ten trial runs [73].

A further alternative is to run the same simulation with a different optimizer. There are many potential optimization methodologies available [23] and the simulation environment should be modular enough to support easily switching the optimizer and executing again. This will serve to alter how the optimizer perturbs the architectures, resulting in a different pattern of exploration across the tradespace.

A final method to confirm that a solution represents a global and not a local optimum is analysis of the results themselves. This can be done by plotting the results of all the simulation runs over the tradespace. If enough runs were performed, Pareto frontiers will be evident in addition to the constraint surfaces. Various local minimums and maximums will be identifiable and it should be possible to directly observe if the optimum selected is global or not. This may or may not be practicable given the complexity of the tradespace. In particular, if many of the constraints and/or contributing analyses were non-differentiable or resulted in large variances in the region of the tradespace that enveloped the identified optimum this may be difficult. Still, it is an acceptable method to investigate and can potentially confirm with certainty that an optimum is global.

These methods of establishing whether or not an optimum is global are not mutually exclusive, and it is advisable to combine them. In particular, first plotting the runs of the initial simulation to directly analyze the tradespace can be useful in informing whether additional runs are necessary, and regions of the tradespace that may be likely to hold a better optimum. This can inform what additional runs should use as an initial input. This is likely to be the most efficient approach for confirmation.

## 2.4 Software Implementation

This section outlines several software tools that can be used in order to implement the various components of this methodology, including the simulation and integration environment, the architecture modeling tool, the contributing analyses applications, and the optimizer.

### 2.4.1 Simulation and Integration Environment

A flexible simulation and integration software suite is the core of this proposed methodology. It must be able to interface with many other model formats, architecture modeling software, and optimization tools. The tool of choice for this research is ModelCenter® by Phoenix Integration, which is marketed as a means of increasing productivity by enabling flexible and automated simulation environments [74]. ModelCenter provides an integration environment and user interface that allows for the linkage and successive execution of different modeling applications. Parameters calculated in one application can be passed to and utilized in the execution of another, generating further outputs, or ingested into an optimizer to restart the cycle. It has been successfully demonstrated for use in architecture optimization in the past [33], and is flexible enough to handle very complicated and detailed cost models covering an array of subsystem options [29].

ModelCenter utilizes flexible Application Programming Interfaces (API's) to allow for the integration of modeling applications using many different data formats. Some common formats are already supported in the base ModelCenter software, such as Microsoft® Excel [75] and MATLAB® by Mathworks [76]. Support for many domain specific software suites is also included in the base package, such as Systems Tool Kit® by Analytical Graphics, Inc [77]. If an application of interest is not covered by an existing API, the environment supports coding of a new interface into the ModelCenter simulation through direct manipulation of APIs [74]. ModelCenter has



sufficient flexibility to support the integration of nearly any model regardless of the application used.

#### *2.4.2 Architecture Modeling Tool*

A suitable architecture modeling tool must be used to develop the architecture descriptions used in this research. SysML was chosen as the architecture modeling language of choice for this research in significant part due to the fact that there are quite a few easily available and established software tools that support it. These tools have a variety of options with extensive features, including automated code and compliance document generation, simulation support, and automated portability to other common engineering views such as DoDAF [78] [79]. For purposes of this research, just SysML architecture modeling with traceability was needed.

Due to availability, previous experience, and wide use, Rational Rhapsody® by IBM was chosen as the architecture modeling tool of choice for this research. It supports SysML architecture descriptions and has a number of automated features to support consistency, traceability, simulation, and testing. [78]

It should be noted that for readability and formatting purposes the diagrams in this document were recreated in Microsoft PowerPoint® after first being generated in Rational Rhapsody.

A potential extension of this research is direct integration of an architecture modeling tool with the simulation environment. This would involve generating executable code from the SysML model, which is possible, although it can be difficult in practice [40]. Given that was not a focus of this research, SysML code execution has not yet been explored. However, it could have utility, and the author is aware of several Government system acquisition programs leveraging Enterprise Architect® by Sparx Systems Pty Ltd. [79] to investigate automated code and document generation

from SysML models in order to enforce consistency, compliance, and improve efficiency of requirement analysis and verification.

#### *2.4.3 Contributing Analyses Applications*

Due to the use of a flexible modeling and simulation environment with API support, the range of potential modeling applications to select for contributing analyses within this methodology is virtually limitless. For the purposes of this research, all contributing analyses were modeled in Excel and MATLAB. This greatly simplified the integration effort since ModelCenter already has built in support for ingesting, manipulating, and executing Excel and MATLAB files within the simulation environment [74].

Exploiting the flexibility inherent in these tools, a sufficient variety of contributing analyses were incorporated to comprehensively test this methodology. In fact, even comparatively simple mathematics software such as Excel can model very detailed and complex contributing analyses, including non-differentiable tradespaces [29]. As integration of more complex and purpose-built modeling applications have been demonstrated in the ModelCenter documentation, it was not seen as a necessary factor to evaluate in this research [74].

#### *2.4.4 Optimizer*

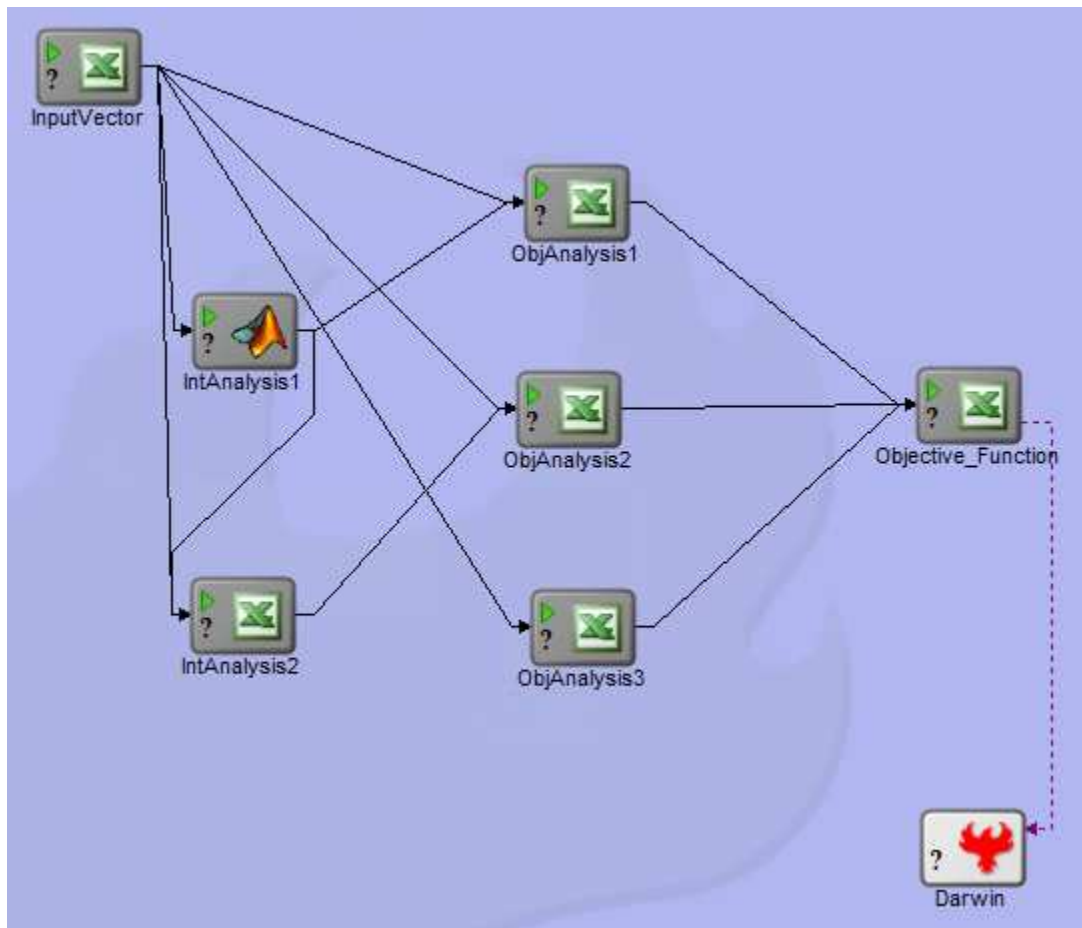
The optimization tool of choice for this research is the Darwin optimizer. This is a built-in genetic algorithm that comes standard within the ModelCenter integration and simulation environment. A genetic algorithm approach was chosen for its inherent flexibility in being able to successfully optimize a wide range of potential objectives including the potential for non-differentiable tradespaces [23].

The Darwin tool has been demonstrated in similar system design studies [33]. Using a built-in and supported optimization tool greatly simplifies the integration effort. The Darwin

algorithm can be seamlessly integrated with the contributing analyses in a modular fashion in the simulation environment.

#### 2.4.5 Setup Overview

A ModelCenter implementation integrating Excel and MATLAB contributing analyses with the Darwin optimizer in the form described by the meta-model in Figure 5 is shown below in Figure 7.



**Figure 7: ModelCenter Example Implementation**

In this ModelCenter implementation, all of the contributing analyses are arranged as prescribed by the scenario in order to calculate the objective function. This objective function then feeds its objective value into the Darwin optimizer which will then compare it to the input vector and modify the input vector. This will run continuously until the simulation is ended either by

meeting a set of pre-defined criteria or when it is stopped by the user. While in this specific implementation all the contributing analyses are in MATLAB and Excel, that is not a constraint on the general applicability of the methodology, and ModelCenter could handle many more contributing analysis formats.

## **2.5 Uncertainty and Sensitivity Analysis**

This section highlights relevant information about uncertainty and sensitivity analysis.

### *2.5.1 Decision Uncertainty*

Decision Uncertainty is the uncertainty due to changing human goals or decision making in the future. This can affect the accuracy of a model if it is predicated on humans operating a certain way and they systemically operate differently, for instance the model assumes humans are under normal conditions and in reality they are under stress and operate differently. This can be handled under other methods to handle systemic uncertainty [80].

A further concern in decision uncertainty is if the model represents the desire of the sponsors of the model, in the case of this research this would be the stakeholders for the architecture selection [80]. The best way to handle this is to enforce using traceable metrics in the objective function so that the decision criteria is clearly documented. If the desires of the stakeholders change, then the objective function should be updated to reflect that. While this won't perfectly capture all stakeholder desires, in the author's experience it is certainly more traceable and clear than current architecture selection methodologies.

### *2.5.2 Subjective Measurement Uncertainty*

Some measures that are common to be included in architecture assessments may rely on subjective expert judgment as the only way to establish a metric, a classic example of which is cybersecurity risk. These types of measures can be handled by this methodology and the goal of

proper uncertainty analysis is not to replace expert judgment with more objective measures, but rather enhance it to allow for validated uncertainty quantification. Specifically, subject matter experts need to provide their assessment in a quantitative fashion rather than in some of the more esoteric methods typically seen (such as ordinal scales involving low, medium, and high or color coded risk matrices). This is typically done through assigning quantified impacts, likelihoods, and a confidence interval [52]. In this manner, an expert judgment can be put into the same form as any other measuring tool, and can be scrutinized and validated in a similar fashion [81]. In fact, expert judgment can be calibrated through feedback and training to provide a more accurate measure of confidence in the assessments, the improvement of which can be quantified and validated through historical analysis. A common example of which is weather forecasters, who tend to be well calibrated in assessing the confidence of their forecasts due to years of repeated experience and feedback [82].

Bayesian methods can also provide a powerful tool for estimating probabilities. For instance, the probability of a cybersecurity breach could be better determined by expert judgment and informed by a penetration test with the original probability updated if that test was positive or not. Bayesian methods also allow to use a node probability table (NPT) to account for various subordinate probabilities. [52]

Beta distributions can also be used to derive probabilities from infrequent events such as cybersecurity breaches. This can be done by either starting with a subjective expert judgment, or with a uniform distribution. In fact, rigorously applying beta distributions likely provides a more accurate probability of cybersecurity breach for firms that have yet to experience a breach due to accounting for that firm just getting lucky in the past. [52]

The Lens method can also be used, which involves building a regression model from a number of experts providing their opinion on a number of scenarios. This is preferable to asking expert judgment directly as it removes expert inconsistency (variation on how the same expert judges the same scenario at different times), which accounts for 21% of expert judgment variation. Expected Opportunity Loss can also be calculated to see if more tests are warranted to collect data to reduce uncertainty. [52]

## CHAPTER 3: CASE STUDY 1: SATELLITE COMMUNICATIONS SYSTEM OF SYSTEMS

This chapter outlines a case study applying the proposed methodology to conduct an architecture selection for a satellite communications SoS.<sup>1</sup>

### 3.1 Case Study 1 Introduction

This case study implements a MBSE methodology to integrate technical and programmatic parameters to solve a best value architecture optimization problem in order to address the needs and constraints of U.S. DoD system acquisition. In this manner, this implementation serves as a tool for improving stakeholder consensus and capturing more thorough traceability for decision factors, while at the same time improving support for variation analysis and iterations on the decision criteria with stakeholders. Unlike many previous multidisciplinary optimizations, this approach is targeted at acquisition activities prior to system design and focuses on the optimization of requirements, including nonfunctional requirements, to be levied on specific systems within the SoS enterprise. It employs architecture-centric parametric analysis to this problem space where concrete system designs do not yet exist.

A SoS is defined as a system whose components are systems in their own right with their own purposes that will continue to serve those purposes if disassembled from the overall SoS [37]. These component systems are managed at least partly for their own purposes rather than the purposes of the SoS. The organization responsible for that SoS capability is often challenged by

---

<sup>1</sup> © [2018] Wiley Periodicals Inc. Re-used with minor grammar and formatting changes, with permission, from M. LaSorda, J. M. Borky and R. M. Sega, "Model-Based Architecture and Programmatic Optimization for Satellite System-of-Systems Architectures," *Systems Engineering*, vol. 21, no. 4, pp. 372-387, 2018. <https://doi.org/10.1002/sys.21444>.

having little or no official budget decision authority over all the constituent systems. These systems commonly have competing requirements or priorities, various technology baselines, and uncoordinated program schedules, complicating any SoS architecture decision [83]. Much of the data that this case study employs is derived from work on a SoS delivering military satellite communications.

MBSE has many potential applications to a space SoS architecture. This case study focuses on improving U.S. DoD satellite acquisition to support Air Force Space Command's (AFSPC) Space Enterprise Vision (SEV) initiative aimed at enhancing the capability of military space systems to operate through a contested environment. Issues include modifying current and planned satellites to make them more resilient to threats, linking acquisition timelines to the emergence of a credible threat, and driving down the development timelines of a military satellite system in order to be more responsive to emergent threats [84]. Critically, this could involve changes to both technical performance parameters as well as acquisition processes. Overall, these acquisition efforts feed directly into a Space Warfighting Construct (SWC) to maintain space superiority, which is the assured ability to operate and survive in space in the face of natural and man-made hazards in the 21st century [85].

The case study begins by investigating optimization techniques through a MBSE approach to shorten the timelines of a satellite system acquisition, while also accounting for cost control and architecture resilience. Shortening satellite acquisition timelines is critical to reduce technology risk from launching satellites with outdated hardware [86]. Resilience is defined as “the ability of an architecture to support the functions necessary for mission success with higher probability, shorter periods of reduced capability, and across a wider range of scenarios, conditions, and threats, in spite of hostile action or adverse conditions [87].” Recent and ongoing space system acquisition



efforts show that these concerns are primarily addressed through trade-offs and optimization among design life, mass, aggregating (combining) or disaggregating capabilities on satellites, additional primary mission capability, and on-board resilience characteristics. This research demonstrates a technique for the predesign phase optimization of a SoS architecture for a given set of technical and programmatic parameters. The vital importance to national security of robust satellite services, together with a history of persistent difficulty in executing and synchronizing acquisition efforts as recently highlighted by U.S. Congressional leadership [88], makes this optimization work very relevant.

MBSE is applied in this work using the Model-Based Systems Architecture Process (MBSAP) [16]. Because of its familiarity, wide use, and software tool support, the System Modeling Language (SysML) is the language of choice [14]. The results are organized using operational, logical, and physical viewpoints [17] as well as the concept of reference architecture (RA), which has been shown to reduce errors, development time, and cost, and which can serve as a construct for trade studies by providing a baseline that facilitates modeling many excursions [18]. Open Systems Architecture (OSA) is emphasized which enables design flexibility [18], allows for maximum component reuse between systems, and is supported under the DoD's "Better Buying Power" initiative [89].

## **3.2 Case Study 2 Research Setup**

### *3.2.1 Develop a Basic Communications Satellite RA*

The first step of this study is to investigate a communications satellite RA in SysML as the basis for optimization studies. Following MBSAP, the first step in that is to document requirements. In this particular scenario, the architecture has three specific capability requirements as shown in Table 2.

**Table 2: CommSat Architecture Requirements**

1) Maintain backwards compatibility with current user terminals
2) Maintain access to +/- 65° latitude around the world
3) Maintain primary services in the Ku frequency band

A communications satellite SoS has three main segments: the space segment, consisting of one or more spacecraft with one or more communications payloads and the support hardware bus, the mission control segment on the ground, and the user segment, which consists of the ground communications terminals employed by users. Notably, the user segment can be managed, acquired, and operated by one or more external organizations that are independent of the organization responsible for the space segment. A primary trade involves allocating capabilities among the three SoS segments. For instance, architecture optimization could include allocating cybersecurity features terrestrially, in space, or in some combination of both, which impacts system costs, usability, and supportability. Another potential trade involves OSA concepts to enhance modularity, loose coupling, and common standards, leading to shorter development timelines. Open interface standards between segments would likely go far to simplify the design. An optimum architecture seeks the best value among these and other potentially competing concerns.

Figures 8 and 9 are respectively a top-level SysML Block Definition Diagram (BDD) for a Communications Satellite Domain Composition and an Internal Block Diagram (IBD) showing a Communications Satellite Operational Context. These diagrams are the foundation for a communications satellite RA that can be validated against existing systems and used to establish the organization and content of the contributing analyses for an optimization study. The RA

captures satellites in orbit and ground and user segments on Earth. This is consistent with a detailed example of how to apply SysML to a space architecture as outlined in Friedenthal and Oster [15].

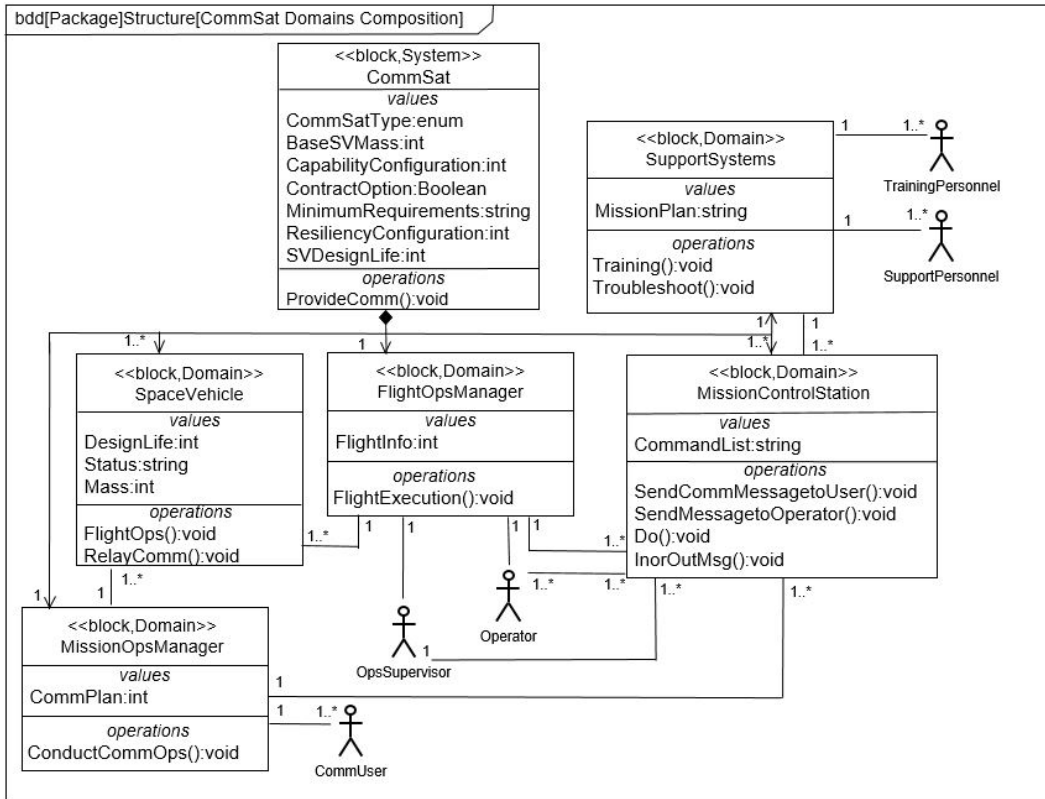
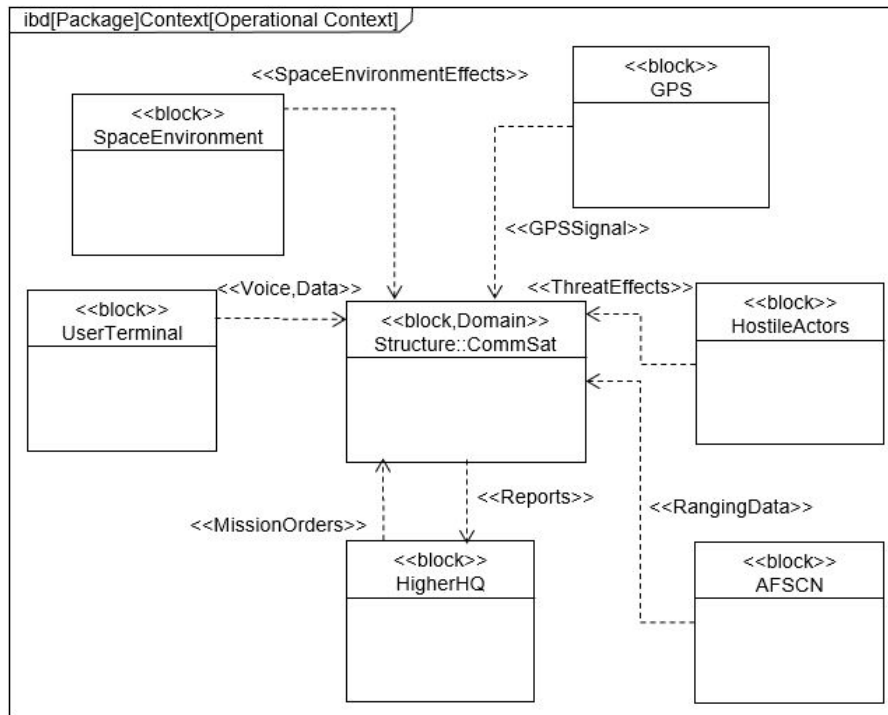


Figure 8: BDD of a Communications Satellite Domain

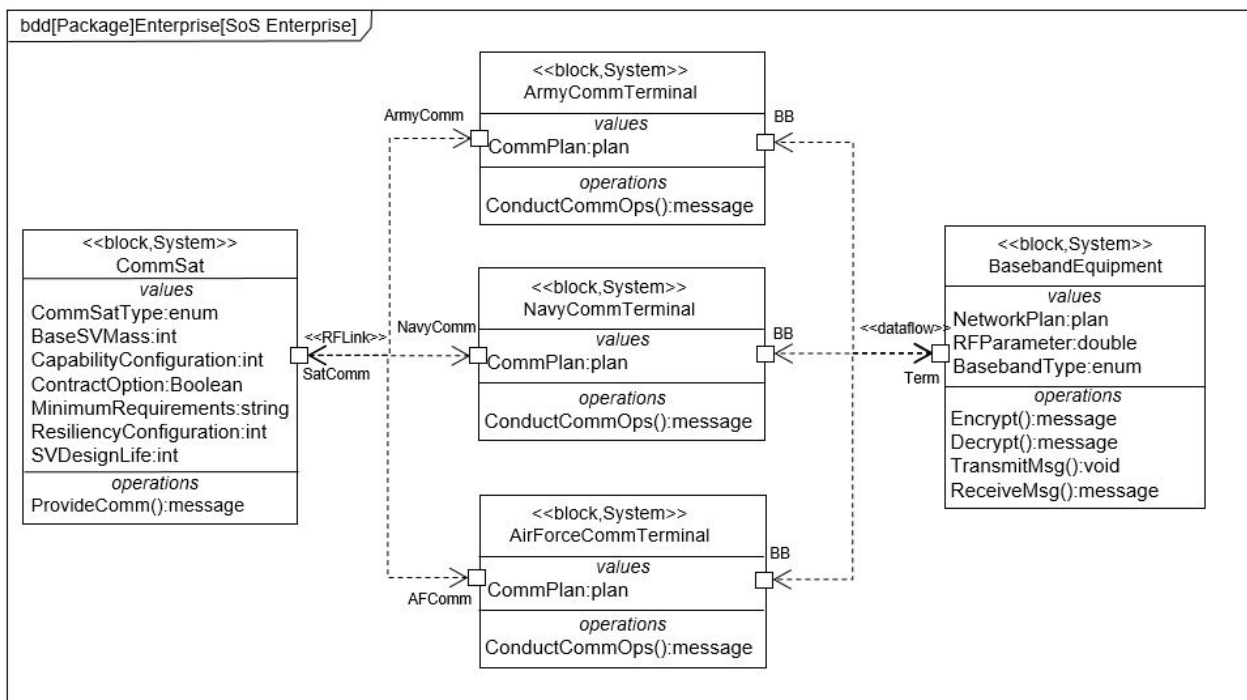


**Figure 9 IBD Communications Satellite Operational Context**

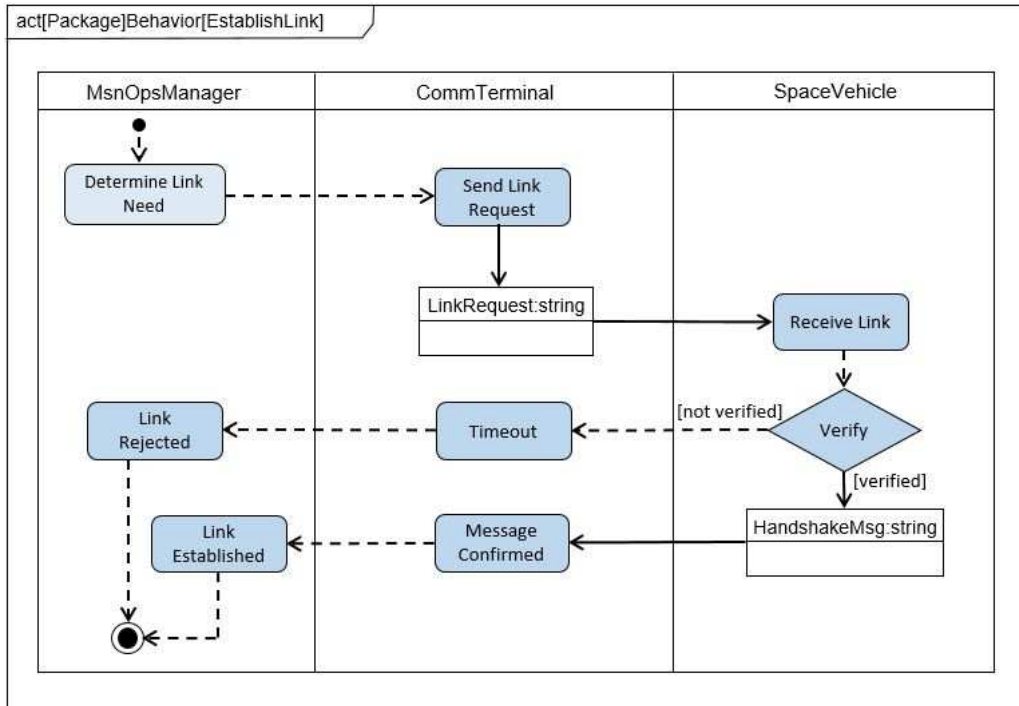
A driving requirement for the space architecture is compatibility with current user terminals that lack the capability to track and hand off between satellites to maintain a connection. This requires satellites in geostationary orbit, maintaining a fixed direction from the point of view of a terrestrial terminal [62]. Another is a requirement to reach high latitudes around the globe, demanding a minimum constellation of four satellites relatively evenly spaced in geostationary orbit in order to reach acceptable look angles above the horizon for the ground terminals. Both these factors are limits on the architecture tradespace. While this served to eliminate evaluating different orbits from the present study, orbit selection is certainly a prime candidate for a satellite architecture optimization [90] and could be integrated into this methodology.

For purposes of this study, it is assumed that each new satellite is potentially a new development effort (and potentially the result of a contract competition) in order to enable the injection of new capability to meet SEV goals. This assumes that proper OSA methodology is

followed to ensure modularity and that the existing ground control system and user terminals will be compatible with any future iteration of the satellite design. This serves to establish the architecture as a SoS, as each satellite iteration, as well as each family of user terminals, is an independent, contributing system to the overall satellite communications enterprise in which various individual satellites, constellations, and terminal types are integrated, the technical dimension of which is the SoS shown in Figure 10. An activity diagram illustrating an interaction of the elements of this SoS is shown in Figure 11.



**Figure 10: System-of-Systems Enterprise Diagram**



**Figure 11: Activity Diagram of Terminal Establish Link Request**

Essentially there is a need to define the general parameters guiding the spacecraft development component of this SoS architecture, such as acquisition strategy, on-board capability, and resilience measures, as well as schedule and cost goals (which in turn drive budget planning).

### 3.2.2 Investigate Contributing Analyses and Data Sources

The second step is to investigate potential contributing analyses and sources of truth data to identify quantifiable metrics as inputs to an optimization problem. One focus is on metrics that can predict schedule timelines and uncertainty, as one of the goals of the SEV is to drive down satellite acquisition timelines [84]. However, the selection of the contributing analyses and data sources is dependent on the problem at hand and is easily tailorable. Any model used in a contributing analyses must be validated in some way and be quantifiable. A number of data sources as identified in section 2.2.4 were considered.

### 3.2.3 Setup of the Optimization Problem

Next the parameters to be optimized must be defined. This should be done with maximum stakeholder participation to arrive at a set of parameters of interest that can be agreed to by all decision makers. In a major defense acquisition, requirements are generated through the very robust Joint Capabilities Integration and Development System (JCIDS), which involves the system operator and other stakeholders in developing reasonable operational requirements for the ensuing acquisition [69]. An oversight process should be required in selecting the optimization parameters and weightings for the proposed method to help support stakeholder buy-in to the final optimization solution. Also, the factors need to be normalized to one another since otherwise factors measuring larger absolute values would dominate the optimization. This normalization is done by dividing the weighting by a normalization factor representing the nominal value for that parameter, which would be selected by the stakeholders.

Since there are numerous potential functions to be optimized, there will typically be multiple objective functions and therefore this can be classified as a multi-objective optimization problem [22]. A multi-objective optimization can be expressed by Equation (1):

$$\text{Eq 1. } g(x) = \min\left[\frac{\alpha_1}{b_1} f_1(x), \frac{\alpha_2}{b_2} f_2(x), \dots, \frac{\alpha_k}{b_k} f_k(x)\right]$$

subject to  $x \in X$ ,

where the integer  $k \geq 2$  is the number of objectives,  $\alpha$  is a weighting factor associated with each objective function,  $b$  is the normalization factor, and  $X$  is the feasible set of decision vectors, typically defined by constraint functions.

The component objective functions are selected based on needs of the stakeholders, with consideration for what can be quantitatively measured and predicted with a calculable confidence. Based on the sources described above, some of the principal potential contributing analyses to be

used as objectives would include cost and schedule measures, along with mature measures of technical performance.

As an example of a notional communications satellite architecture that seeks to meet SEV goals and can be used to explore the methodology, four contributing analyses were selected: time to first launch from contract authority to proceed, annual architecture cost, threat effectiveness, and excess capability beyond threshold.

Time to launch from contract authority to proceed is defined as the time from when contractual direction is given to build a satellite, to when that satellite is launched. This contributing analysis could be seen as valuable to implementing the SEV which has stressed shorter satellite acquisition timelines to respond to emerging adversary threats and the challenges of technology obsolescence. The function for this could also reasonably be expected to be quantified with some predictive relationship as there are numerous historical examples from previous satellite programs to construct an estimating relationship. This function should be minimized. It is normalized by dividing the calculated value by 60 months, which corresponds to a goal of a launch within five years.

Annual architecture cost (cost of a SoS conforming to a given alternative architecture) was chosen because system affordability is always a concern. In particular, an annual cost was selected versus a per system cost to enable investigation of the design life tradespace. For instance, it might be beneficial to launch cheaper satellites more often versus more expensive satellites less frequently. This evaluates the system on how much it costs to continuously maintain service, which includes launching satellites to replenish failed units. This function is calculated in \$M/yr and should also be minimized. It is normalized by dividing the calculated value by 600 which corresponds to a nominal objective of \$600M for the annual architecture cost based on comparison



to similar constellations. Furthermore this annual architecture cost function will have a maximum constraint of 750 which was selected as an upper limit to simulate the fiscal reality of having to make this architecture fit within a departmental budget.

Improving space systems' resilience to threats (including hostile actors and the space environment) is a large focus in the SEV, which is why threat effectiveness is a selected function of interest. It might appear nebulous to quantify, but there are numerous predictive models for various survivability and other performance metrics, such as radiation hardening, that can be folded into a threat effectiveness metric. In this way mature measures of technical performance that could be quantified, and are acceptable to the stakeholders, would serve as a surrogate for evaluating resilience and be folded into a measure of threat effectiveness. This function should be minimized.

Lastly, capability is always a concern. While capability can be measured in many ways, and defined under operational conditions, it is often defined for DoD acquisitions through KPPs [60]. These KPPs have to be measurable and therefore are quantifiable metrics. The capability tradespace for a Government system is defined as the range between the threshold KPP (the minimum required of the system) and the objective KPP (the desired level of capability for the system). For a communications satellite system, there are numerous technical models of how a satellite's capability is tied to predictive metrics. For instance, size of the effective antenna will drive gain which in turn helps establish link margin and supportable data rate [62]. This function is defined as the difference between the architecture's capability and the objective value, therefore minimizing this parameter is desirable.

Weighting factors for the analyses of interest will have to be decided among all the stakeholders. For this research, the weighting factors selected are shown in Table 3. These weighting factors can be varied to explore alternative scenarios.

**Table 3: Case Study 1 Objective Function Weighting Factors for Parameters of Interest**

Parameter of Interest	Weighting
Time to First Launch ( $f_1$ )	0.4
Annual Architecture Cost ( $f_2$ )	0.3
Threat Effectiveness ( $f_3$ )	0.2
Capability to Objective ( $f_4$ )	0.1

Overall this would result in the multiobjective optimization shown in Equation (2):

$$\text{Eq 2. } g(x) = \min \left[ \frac{.4}{60} f_1(x) + \frac{.3}{600} f_2(x) + .2f_3(x) + .1f_4(x) \right]$$

subject to:

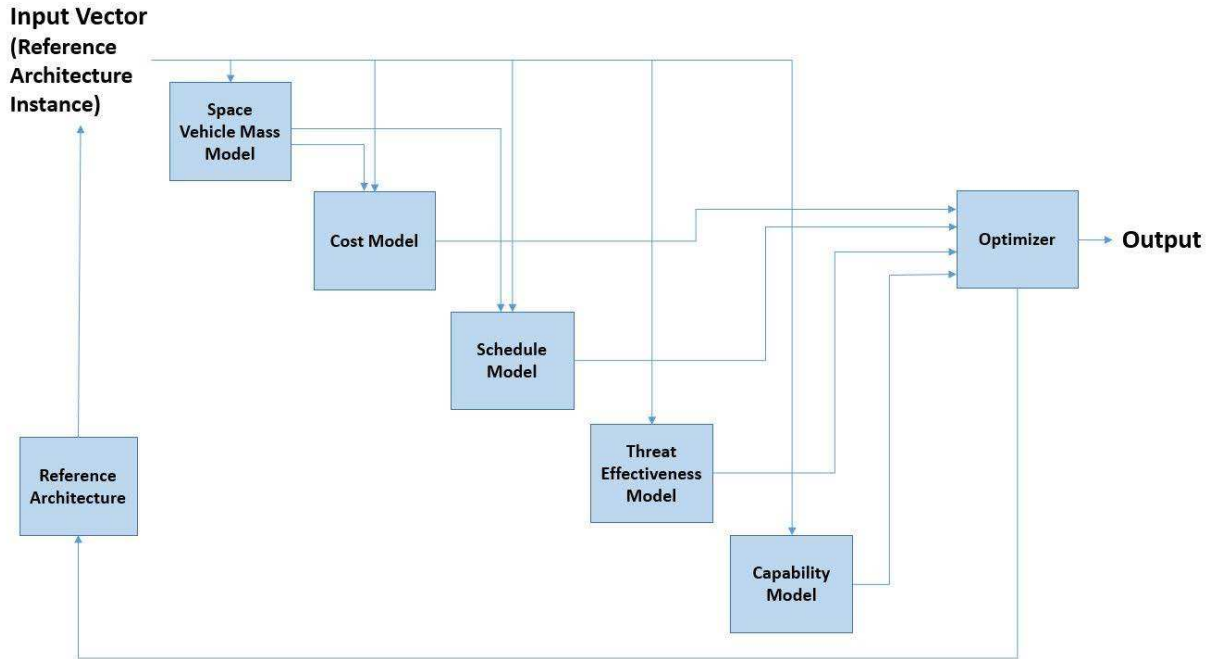
$$f_1(x), f_2(x), f_3(x), f_4(x) \geq 0$$

$$f_2(x) \leq 750$$

Other than  $f_2(x)$ , the upper limits of the contributing objective analyses are defined by the constraints on their input vectors.

### 3.2.4 MBSE and Optimization Integration

The next step in the case study is to integrate the SysML architecture models with the optimization models. A meta-model of the relationships between contributing analyses and the optimizer is shown in Figure 12. In this diagram the optimizer takes the outputs of the contributing analyses into an objective function and outputs a result. It then tries to improve the output of the objective function by changing the current instance of the RA.



**Figure 12: Case Study 1 Meta-model of Contributing Analyses**

Continuing with the methodology, a Variability BDD is defined describing the trade space to be investigated that can contribute to the objective function. Figure 13 shows this Variability BDD in SysML. The main trades identified are longer or shorter design life durations for the satellites, variable satellite capability, variable satellite resilience, and an option to execute the development as a contract option versus a newly competed contract. These options were selected because they are reasonable variables to control in the acquisition of a satellite system. They also give a range of example trades that could be modeled with this process. The selection of trade options would be highly adaptable to the study at hand.

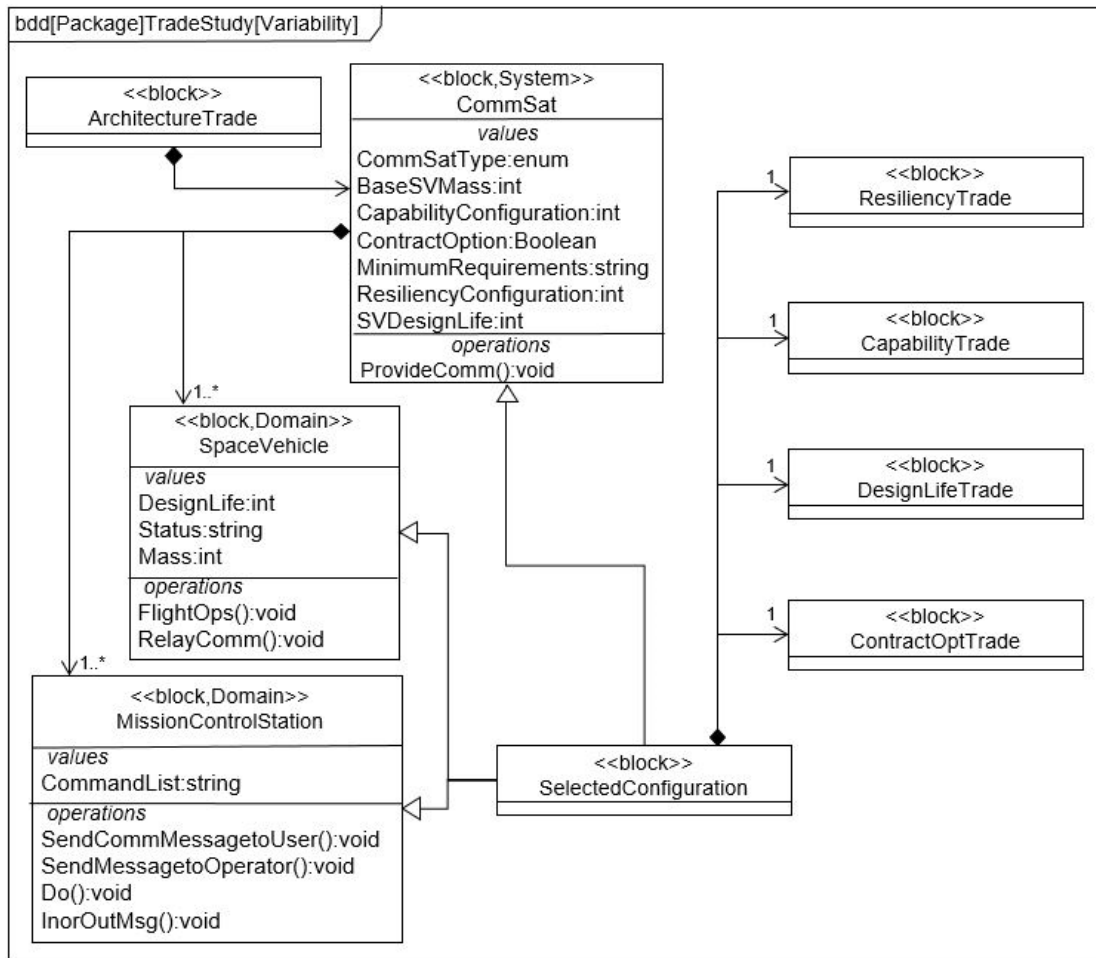


Figure 13: Case Study 1 Variability BDD in SysML

### 3.2.5 Contributing Objective Analyses Selection

Based on the above sources, and the trades defined in the Variability BDD, several relationships were selected as contributing objective analyses. Some of these are notional for this research to protect information that is not publically releasable, but all are based on real-world relationships to the maximum extent possible. The intent is to demonstrate how this technique could be applied to a real-world problem, and these notional relationships serve as realistic surrogates for real-world contributing analyses. There are four models that each correspond to a contributing objective function, and a fifth model, *Space Vehicle Mass*, that is a necessary intermediate step for several of the other models.

### 3.2.5.1 Time to First Launch Contributing Analysis

For the *Time to First Launch* model, which outlines the length of time in months required for a new satellite development, the relationship has been modeled by Equation (3):

$$\text{Eq 3. } \textit{Time to First Launch}(\textit{mos}) = 7.9 + 0.69 (m * 2.205)^{0.408} * DL^{0.179} + 11.8MT - 7.1Opt$$

subject to:

$$0 \leq m \leq 10000$$

$$0 \leq DL \leq 480$$

$$0 \leq MT \leq 5 \text{ and integer}$$

$$Opt = 0 \text{ or } 1$$

where  $m$  is the projected mass of the space vehicle in kilograms,  $DL$  is the design life of the spacecraft in months,  $MT$  is the number of mission types on the spacecraft, and  $Opt$  is a Boolean variable for whether the satellite is being executed as a contract option or new contract competition (1 corresponds to a contract option). This function is a schedule estimating relationship (SER) based on historical data commonly used for Government spacecraft [91]. Investigating mission types and aggregation of multiple missions on one satellite was not investigated in this study ( $MT$  was set to 1 and not varied) but it is a focus for future work.

### 3.2.5.2 Annual Architecture Cost Contributing Analysis

An *Annual Architecture Cost* model is less straightforward. The model selected was developed to account for regular launches utilizing known cost estimating relationships (CERs) which are described below. For a new development program CER, development costs can be modeled at a high level by Equation (4):

$$\text{Eq 4. } \textit{Cost}(\$M) = 1.879 (m * 2.205)^{0.6889}$$

subject to:

$$0 \leq m \leq 10000$$

where  $m$  is the projected mass of an individual space vehicle in kilograms. For this CER it is assumed all design considerations such as design life, resilience, and capability are already factored into the mass estimate. This CER is from the Unmanned Satellite Cost Model (USCM) calculated directly from a study of 16 Government satellite development efforts [92]. This is a regression model with a Pearson's  $R^2$  of 0.3878. The model is publically releasable, but the actual data contains proprietary information and is not; however, the information was available to the author from the U.S. Air Force Space and Missile Systems Center.

To convert this CER into an annual cost for this particular architecture, it was multiplied by a factor to average the satellite cost per year at a rate to replenish the needed number of satellites on orbit (4 for around the world coverage to meet requirements for this example constellation) at their design life. This would equate to the per satellite cost multiplied by  $DL$  (which is calculated in months) divided by 48.

Other components of the *Annual Architecture Cost* include a base annual cost to cover sustainment and regular system upgrades for the mission control segment. This was selected to be \$50M, based on the author's observations of large Government satellite contracts; however, in a real scenario there would be historic data to estimate sustainment costs. It is feasible to break out the cost of the mission control segment independently of the space vehicle such as this if proper OSA methodology is being followed to ensure modularity [16]. A factor was added using the  $Opt$  variable to show cost savings of approximately 10% due to competition. This factor would be derived from market research and is very dependent on the situation at hand. While savings over 30% can be realized through contract competition [93], in more highly specialized fields the

margin is often narrower due to limited competitors and 10% was assessed to be the best estimate for savings that could be expected from competition in this scenario. The full *Annual Architecture Cost* model is shown in Equation (5):

$$\text{Eq 5.} \quad \text{Annual Architecture Cost} \left( \frac{\$M}{\text{yr}} \right) = 50 + 1.879 (m * 2.205)^{0.6889} * (0.9 + 0.1 * Opt) / \left( \frac{DL}{48} \right)$$

subject to:

$$0 \leq m \leq 10000$$

$$0 \leq DL \leq 480$$

$$Opt = 0 \text{ or } 1$$

where  $m$  is the mass of the spacecraft in kilograms,  $DL$  is design life in months, and  $Opt$  is a Boolean variable where 1 corresponds to the satellite being built as a contract option and 0 as a new contract competition.

### 3.2.5.3 Threat Effectiveness Contributing Analysis

The *Threat Effectiveness* model accounts for the probability that a satellite can be prevented from operating properly by an adversary's actions or space environment effects. This model is the author's creation based on experience in spacecraft design and it demonstrates how quantified performance against threats on a normalized scale can be integrated into architecture optimization. In the architecture, there are three "Resilience Configuration" metrics that each mitigate a different threat, and each has a different effect on the mass of the spacecraft. An example Resilience Configuration metric could account for radiation shielding to protect against unforeseen radiation-based threats such as solar flares, which would have a mass impact proportional to the overall mass of the space vehicle and the amount of shielding used. While these relationships are notional, they are based on experience and were selected to show a range of plausible options. They are linked

in the *Space Vehicle Mass* model to have varying impacts on the final space vehicle mass, which is a realistic consideration to add additional capability to a satellite [62]. The largest threat effectiveness of the three is taken as the overall threat effectiveness since the resilience of the spacecraft is assumed in this model to be measured on the worst potential risk a threat poses to the spacecraft to address the spirit of the SEV. The *Threat Effectiveness* model is shown in Equation (6):

$$\text{Eq 6.} \quad \textit{Threat Effectiveness} = \max_{i=1}^n |(1 - RC_i)|$$

subject to:

$$0 \leq RC_i \leq 1$$

where  $RC_i$  is the resilience configuration and  $n$  is the number of different resilience measures being considered. For this specific case three were considered, identified as  $RC_a$ ,  $RC_b$ , and  $RC_c$ . They all are normalized to have a value between 0 and 1 and were selected as representative of how different resilience features could impact space vehicle mass.

Expanding on the example above, having no additional radiation shielding would correspond to a  $RC_i$  score of 0 and having the maximum radiation shielding desirable would correspond to 1. The  $RC_i$  score is then translated to have an impact on the final mass of the spacecraft in the *Space Vehicle Mass* model.

#### 3.2.5.4 Capability to Objective Contributing Analysis

The *Capability to Objective* model measures the difference in actual capability to the objective capability value. Minimizing this shows increasing capability as the space vehicle's performance gets closer to the objective. This is dependent on the capability configuration of the spacecraft, which is a measure of additional capability built into the spacecraft and is linked to the Space Vehicle Mass model to have an effect on the final mass of the spacecraft. Similar to the



Threat Effectiveness model, this is a notional model but is grounded in the reality that additional capability will lead to more mass [62]. The overall capability to objective model is shown in Equation (7):

$$\text{Eq 7.} \quad \textit{Capability to Objective} = 1 - CC$$

subject to:

$$0 \leq CC \leq 1$$

where  $CC$  is the capability configuration of the spacecraft and is normalized to have a value between 0 and 1. For purposes of this study, the objective value was taken as 20% greater than the threshold capability, which could be defined by a KPP such as bandwidth. As an example, if a bandwidth of 5 GHz was the minimum threshold capability as defined by the stakeholders, it would equal a  $CC$  score of 0. The corresponding objective bandwidth then would be 6 GHz and correspond to a  $CC$  score of 1. Throughput requirements such as this can be typical in communications satellite programs of this kind, based on a review of historical program data.

### 3.2.5.5 Space Vehicle Mass Contributing Analysis

A contributing analysis that is not a direct input to the objective function but is a necessary intermediate step is *Space Vehicle Mass*. Both *Time to First Launch* and *Annual Architecture Cost* models are dependent on *Space Vehicle Mass* while parameters necessary to calculate *Threat Effectiveness* and *Capability to Objective* are inputs to *Space Vehicle Mass*. In this sense, the *Space Vehicle Mass* contributing analyses is where many of the tradeoffs happen. The *Space Vehicle Mass* model is shown in Equation (8):

$$\text{Eq 8.} \quad \textit{Space Vehicle Mass}(\text{kg}) = \left[ \left( Bm * 1.01^{\frac{DL}{12}} \right) * (1 + 0.2CC)^3 + 181.4 * RC_a \right] * \\ (1 + 0.15RC_b) * 1.3^{RC_c}$$

subject to:

$$0 \leq Bm \leq 10000$$

$$0 \leq DL \leq 480$$

$$0 \leq RC_i \leq 1$$

$$0 \leq CC \leq 1$$

where  $Bm$  is the base mass of the space vehicle in kilograms (set to 4082 initially based on the author's experience with large Government communications satellites, but could be changed depending on the space vehicle of interest),  $DL$  is design life in months,  $RC_a$ ,  $RC_b$ , and  $RC_c$  are Resilience Configurations A, B, and C respectively with a value between 0 and 1, and  $CC$  is Capability Configuration with a value between 0 and 1. The model is complex, and is not derived from a regression but rather from a series of analytical relationships which are explained below:

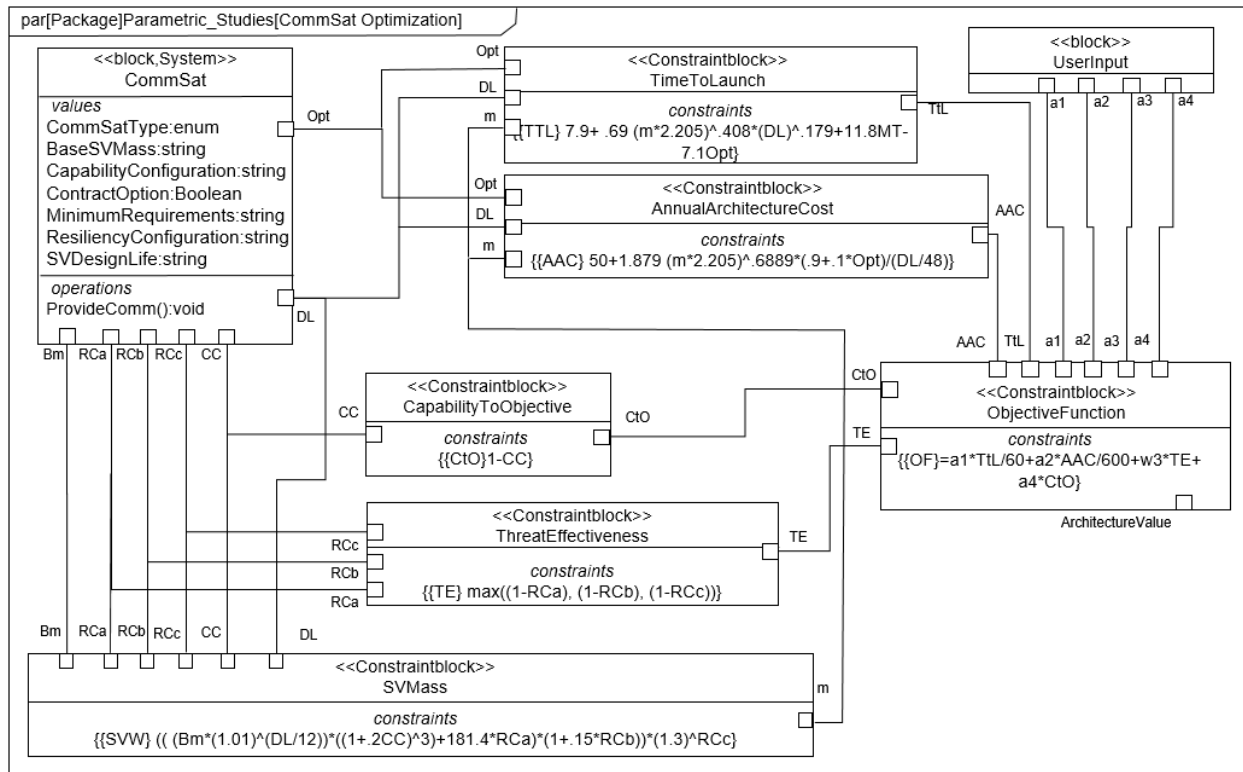
- Design life increases mass as more equipment is needed to ensure redundancy and engineering margin against the naturally degrading environment of space. This contribution was derived from known engineering estimates of annual solar panel degradation (a main life limiting constraint on spacecraft) of 3.75% per year and an electrical power subsystem representing 30% of the mass of the space vehicle [62].
- Capability Configuration exponentially increases the mass based on the physical relationship that linearly increasing the aperture (and therefore capabilities) of the payload antennae will have a cubic effect on the final mass of the payload, which will then proportionally increase the final mass of the space vehicle [62]. While there are other factors that contribute to a communications satellite capacity that may not have a cubic effect on mass, aperture size was taken as a representative design variable. This technique is extensible to more complicated scenarios with multiple design variables [29].

- Resilience Configuration A linearly increases mass with a maximum addition of 181.4 kg to the base configuration of the satellite. This is notional but would be analogous to adding a specific additional hardware package on the spacecraft in the form of a secondary payload that provides a capability linearly scaling with mass.
- Resilience Configuration B increases the base mass and any added hardware from  $RC_a$  and  $CC$  by up to 15%. This is also notional but can be seen as analogous to adding hardware that would have a mass impact proportional to the entire spacecraft mass, an example of which could be radiation hardening.
- Resilience Configuration C increases the base mass and mass for all other contributions by a 1.3 factor exponentially. This too is notional but represents adding a capability that has different returns for additional mass, for example, additional fuel for contingencies.

While all these models contain assumptions and notional components, the intent is to display the breadth of applicability with this technique to numerous potential contributing analyses. When performing a real-world optimization, many of the contributing analyses will likely be proprietary, not publically releasable, or classified, but parameter values will be well established for the particular optimization analysis.

### *3.2.6 Optimization Software Implementation*

The next step is to convert the Variability BDD into a SysML Parametric Diagram through the selected contributing analyses to outline the quantified relationships in this architecture. This will take advantage of the relationships discovered during the investigation of truth data phase. Figure 14 shows this diagram.



**Figure 14: Case Study 1 Parametric Diagram in SysML**

Then this Parametric Diagram can be converted into an executable model that can be linked with an optimization algorithm. ModelCenter by Phoenix Integration is one platform that provides an integration workspace that enables the linkage of contributing analyses with an optimization function. The above Parametric Diagram was converted into a ModelCenter implementation. The inputs, contributing analyses, and the objective function were built in either Microsoft Excel or MathWorks MATLAB. As noted in Chapter 2, ModelCenter is flexible enough to support integration of nearly any model regardless of the tool used.

A non-gradient-based optimization is necessary since elements of the objective function are not differentiable [90]. The Darwin genetic algorithm was selected as the optimization algorithm due to use in previous demonstrations of similar techniques [33]. The overall structure of the ModelCenter setup is shown in Figure 15. Additionally, after the main simulation was

complete, several shorter runs were performed to ensure consistency and verify that the converged minimum represents a global and not a local minimum.

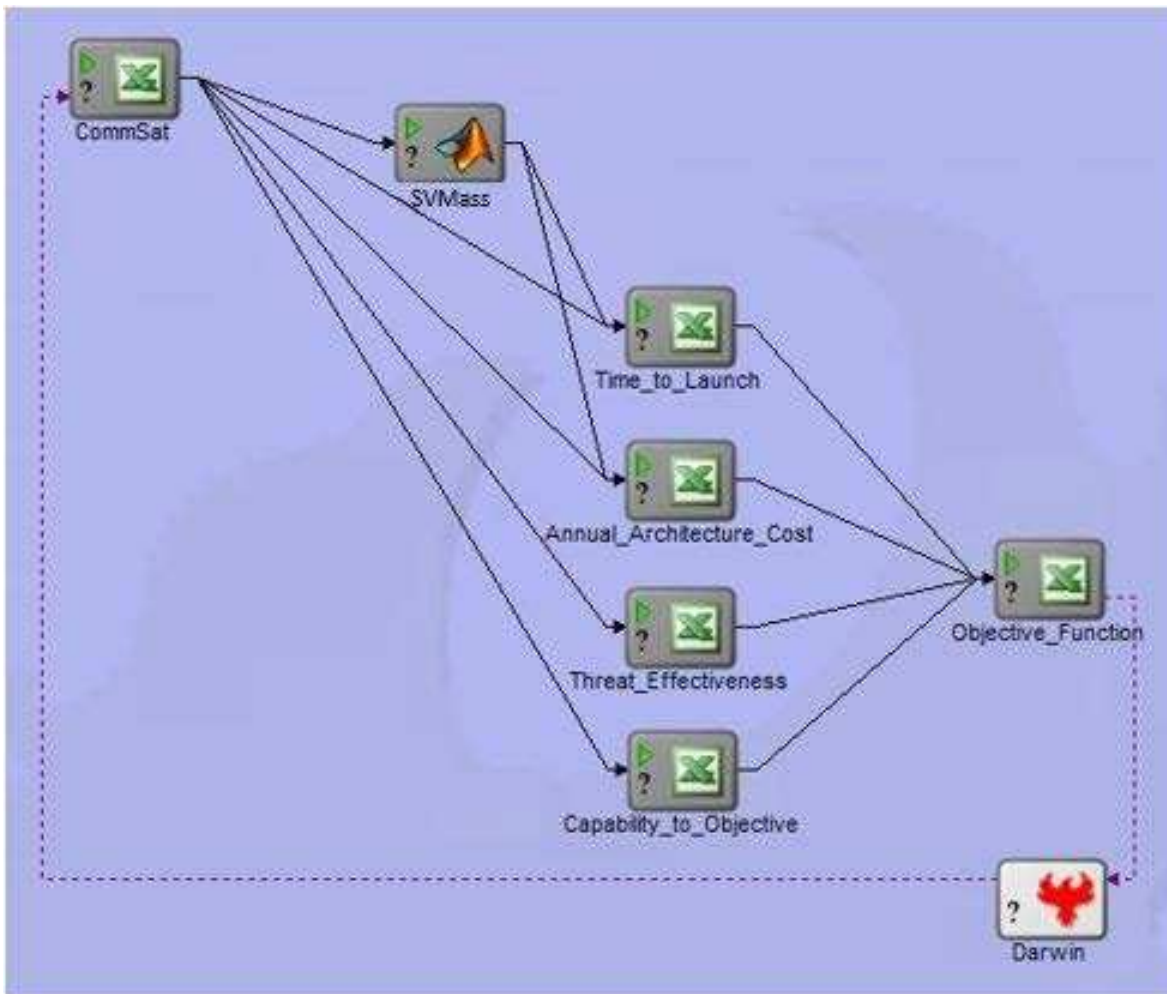


Figure 15: Case Study 1 ModelCenter Implementation of Overall Objective Function

### 3.3 Case Study 1 Results

#### 3.3.1 Simulation Output

The simulation was successful in optimizing a satellite architecture both technically and programmatically. Once exercised through the Darwin algorithm, the simulation converged on an optimum solution per Equation (2) after evaluating 11,853 potential architectures. Optimization was ended after 11,853 runs as there was no longer any significant variation in the results, showing convergence of the optimization that is acceptably close to the theoretical minimum. Additionally

three smaller optimizations were conducted which validated consistency in the results and demonstrated that this was a global optimum and not a local optimum. During the 11,853 runs of the principal simulation, the objective function value varied between 0.946 at the minimum and 3.421 at the maximum. The annual architecture cost varied from \$257M up to the maximum limit with \$387M corresponding to the optimum architecture. The selected optimum solution is shown in Table 4. Of particular note is the relatively high design life that resulted.

**Table 4: Case Study 1 Optimum Architecture Results**

<b>Parameter</b>	<b>Value with Optimum Architecture</b>
Objective	0.9457
Input <i>DL</i>	199.2 months
Input <i>RC<sub>a</sub></i>	0.752
Input <i>RC<sub>b</sub></i>	0.752
Input <i>RC<sub>c</sub></i>	0.751
Input <i>CC</i>	0
Input <i>Opt</i>	1
<i>Annual Architecture Cost</i>	\$387,801,000
<i>Space Vehicle Mass</i>	6710 kg

ModelCenter also conducts sensitivity analysis on the data. Influence factors (partial derivatives of the objective result with respect to the input variable) for the inputs on the objective about the optimum solution are shown in Table 5. Design life had a large negative influence and capability configuration had a large positive influence.

**Table 5: Case Study 1 Influence Factors for Objective**

<b>Parameter</b>	<b>Influence Factor</b>
Input $DL$	-0.573
Input $RC_a$	-0.188
Input $RC_b$	-0.143
Input $RC_c$	-0.140
Input $CC$	0.369
Input $Opt$	-0.270

Influence factors for Annual Architecture Cost were also calculated to provide greater insight into the driving factors in the optimization, and these are shown in Table 6.

**Table 6: Case Study 1 Influence Factors for Annual Architecture Cost**

<b>Parameter</b>	<b>Influence Factor</b>
Input $DL$	-0.716
Input $CC$	0.105

### 3.3.2 Preliminary Validation

This approach to the SoS optimization challenge requires validation as a prerequisite for adoption in the system acquisition process. A promising source of validation involves comparisons of the results to historical program data. Earlier sections of this paper have identified the use of historical data to define the relations used in the analysis. In addition, the optimization results have been compared to existing communications satellite systems that have resulted from conventional experience-based design refinement, and the overall agreement is good. For instance, the increase to space vehicle design life is in line with military satellite communications trends, which have

consistently lengthened to exceed 14 years [94]. The optimum solution identified in the simulation had a design life of 16.6 years. Similarly the trend of increasing mass and additional capability is validated by historical data. A legacy DoD geosynchronous communications satellite, MILSTAR, had an approximate space vehicle mass of 4535 kg while its more capable follow-on DoD program, AEHF, had an approximate mass of 6168 kg [95]. The simulation space vehicles, which had additional resilience capability, had a mass of 6710 kg at the optimum solution.

An additional attempt to provide validation involved comparing these results to other satellite architecture optimization studies. It proved difficult to find comparable real-world analyses that included factors such as shorter design life and increased resilience. Most academic studies of space system optimization focus on life cycle cost to the exclusion of other factors. However, trades in the results were consistent when similar corresponding trades appeared in other studies. For instance, Gross and Rudolph [96] conducted a sensitivity analysis of a rule based design approach for an earth sensing satellite system. The mass of the tracking, telemetry, and control subsystem, which communicates to the ground, was highly dependent on the data rates it was required to support. This is consistent with this analysis, in which additional communications capability impacted payload and overall space vehicle mass, resulting in an influence factor of 0.369 for the CC parameter, the only parameter that had a positive correlation with our objective function. The additional mass required for increased capability did not warrant the extra cost as selected by the stakeholders for the weightings given in the objective.

The validation strategy will continue to evolve and be used as an integral part of ongoing work to model a wider range of scenarios, analyze sensitivities of the optimization result to various analysis factors, and assess the propagation of uncertainties in the data through the analysis. This preliminary validation, while incomplete, provides confidence to continue exploring these efforts.



### 3.4 Case Study 1 Discussion

A full uncertainty analysis will require additional simulations with a range of scenarios and is a subject for the next phase of this research in upcoming case studies. Similarly, more data and a more in-depth sensitivity analysis is needed to increase confidence in the validity of the model. However, this initial optimization study research leads to some key conclusions, and is the basis for continuing work. Specific findings include the following:

- Implementing the SEV will be complicated in terms of design trades. Design life, which would typically be reduced when acquiring smaller, cheaper satellites to implement SEV goals of shorter development timelines, is very negatively correlated with the overall objective and with *Annual Architecture Cost*. In fact, among the input parameters it was the greatest influencer of the optimum, likely due to this direct relationship with higher cost associated with more rapid satellite replenishment. As a result of this negative correlation and the desire to minimize the objective function, the optimum solution actually selected a design life higher than the initial set points.
- Meeting the capability threshold is very stressing under these parameters. The *Capability to Objective* measure was 1 at the optimum solution, meaning that no additional capability above threshold was recommended. In fact, the capability configuration input was the second largest influencer of the objective function after design life, although this was a positive correlation. This was likely due to the significant effect additional capability has on space vehicle mass and the resulting impact *Annual Architecture Cost* and *Time to First Launch*. Since this was a positive correlation, capability configuration was minimized with the objective function, all the way to its lower limit of 0.

- Resilience as defined in this study is also expensive in terms of mass. Architecture solutions around the objective solution implemented all resilience features to a significant degree, which raised the space vehicle mass considerably. All the Resilience Configuration metrics were approximately equal in the optimum solution, which is reasonable since only the lowest (worst) of the three would affect the objective function while increasing resilience increased mass.
- Performing acquisitions utilizing contract options rather than competitions was the preferred solution, likely due to its effect on reducing development timelines. This could well influence the contracting approach for communication satellite acquisition. While current DoD policy mandates competition in most instances [93], modeling techniques such as this could identify situations in which waiving that requirement is in the best interest of the stakeholders.

The parameters of this simulation drive a solution that leads to larger, more capable, longer lasting, more resilient, but more expensive satellites. This seems to be at least partially opposed to the spirit of the SEV, one of the goals of which was more rapid satellite acquisition. This suggests the objective function should be scrutinized by stakeholders. The results also suggest that the feasibility of the SEV with the current acquisition approach is debatable. This might lead to a greater urgency for other acquisition constructs, such as a more modular method of building space vehicles, perhaps a standard bus approach, that would better support OSA and therefore more rapid development timelines. Similarly pursuing mission disaggregation concepts to reduce individual vehicle complexity might increase in attractiveness. These are among the excursions from this initial scenario to be examined in some of the next case studies.

### 3.5 Case Study 1 Conclusion

A technique to apply model-based architecture and programmatic optimization to a satellite communications SoS acquisition was demonstrated, with a focus on implementing the AFSPC SEV. This initial exploratory study provided some useful results which appear reasonable in light of the experience of the space acquisition community, such as a tendency to drive towards more capable but more expensive solutions.

Overall, a new implementation of an architecture-centric methodology was developed to conduct an architecture and acquisition strategy selection trade study in the early phases of military communications satellite acquisition. This new process, an “Architecture and Programmatic Optimization Process” is shown in Figure 16.

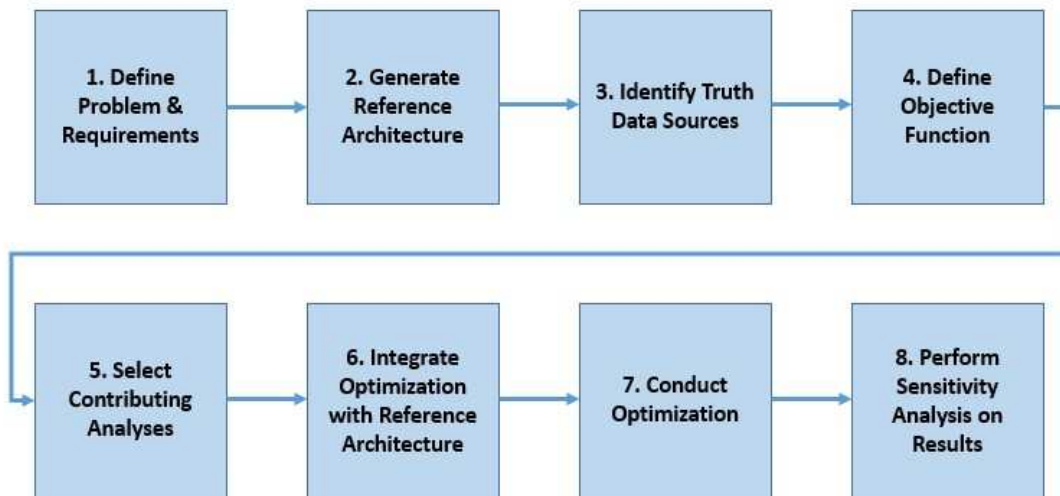


Figure 16: Case Study 1 Overall Architecture and Programmatic Optimization Process

This process is important for establishing an approach that not only optimizes a specific technical architecture, but also holistically includes programmatic concerns that are critical to a program's overall success in meeting cost, schedule, and performance goals. This process will

allow for a better informed acquisition decision by highlighting key trades and results among the candidate solutions.

A more ambitious goal is to better understand the uncertainty involved with this process so that a decision maker will trust the optimization to actually make architecture decisions. Critically, an understanding of the optimum solution's robustness in the face of changing circumstances (all too common with lengthy space acquisitions) will need to be understood. These areas will be a focus for upcoming research.

## CHAPTER 4: CASE STUDY 2: REMOTE SENSING

This chapter outlines a case study applying the proposed methodology to conduct an Analysis of Alternatives (AoA) for a remote sensing satellite constellation and compares the new process to the conventional AoA process.<sup>2</sup>

### 4.1 Case Study 2 Introduction

#### 4.1.1 Current Analysis of Alternatives Process

U.S. DoD acquisitions undergo an AoA evaluation early in the acquisition lifecycle in order to help decision makers understand the tradespace for new materiel solutions to satisfy an operational need. This AoA process is an analytical comparison of the operational effectiveness, suitability, and life-cycle cost of alternatives that satisfy established capability needs. Successful completion of AoAs must be certified to Congress for Major Defense Acquisition Programs (MDAPs). Per current DoD guidance, cost analysis is performed separately from measure of effectiveness and performance analysis [2].

Current AoAs are usually long, expensive processes. Typical recent Air Force AoAs had an average cost of \$15M and took 21 months to complete [97]. This is considered a worthwhile investment as the resultant decisions could impact many billions of dollars of Government acquisitions.

#### 4.1.2 Opportunities for Improvement

Within the author's defense system acquisition experiences, the AoA process has opportunities for improvement, in particular with how they enable a final decision on a materiel

---

<sup>2</sup> © [2018] IEEE. Re-used with minor grammar and formatting changes, with permission, from M. LaSorda, J. Borcky and R. Sega, "Model-Based Architecture Optimization for Major Acquisition Analysis of Alternatives," in *2018 IEEE Aerospace Conference*, Big Sky, 2018. <https://ieeexplore.ieee.org/document/8396526/>

solution. While many different criteria are thoroughly analyzed within the current AoA process, all too often how that influences a final decision is unclear as the results for the various criteria are somewhat subjectively compared at senior-level decision meetings. Frequently during many evaluations, the architecture corresponding to the lowest cost for acceptable performance is selected as cost is easily quantifiable and therefore stakeholders that push for lowest cost can often make the most persuasive argument. The result may not actually be the best value for the stakeholders.

In a resource-constrained environment, it is critical to provide the greatest capability for the cost. Very often, the lowest cost to meet the minimum requirements isn't at that value point. The U.S. DoD acquisition community has conducted an initiative known as "Better Buying Power" since 2010 focused on improving the value the DoD achieves for each dollar spent. The latest version, "BBP 3.0" specifically advocated for identifying better objective measures of value for DoD and industry alike [89].

By exploiting new MBSE techniques, this research seeks to demonstrate a better, more rigorous, model-based analytical approach to objectively measure and select best value from an architecture trade study, specifically an AoA. The results can also be applied more generally to architecture optimization in many system categories.

#### *4.1.3 Problem Statement*

Typically in a Space Acquisition AoA, a handful of driving design characteristics are selected to be varied, such as orbit, aggregating (combining) multiple missions on one physical satellite, and varying capabilities. A handful of candidate architectures then are generated representing the boundary conditions and are independently evaluated on how they perform on

various performance measures, risk, and cost per DoD guidance. In practice, these solutions are then debated by various groups of stakeholders until concurrence is eventually realized.

Current U.S. DoD AoAs, especially for DoD space acquisitions, often struggle with being able to convey the relative importance of additional capability and other measures against cost, which drives the decision toward the cheapest solution that meets the minimum requirements rather than selecting what may be the best value. A contributing factor to this is a lack of clear guidance on the relative importance of the other measures of success, leading to multiple stakeholders having multiple opinions. Cost tends to be a major forcing function, so that minimizing cost tends to result when concurrence can't be reached on what provides the best total value.

Given the push to provide clear, objective definitions of “best value” for DoD systems rather than just minimizing cost [89], this research considers potential modifications to the current AoA standard process in pursuit of this goal. This case study outlines a modified approach to leverage MBSE in AoAs in order to arrive at a best value solution. This new approach is then compared to the current standard AoA process and the results appear promising.

#### *4.1.4 New AoA Approach Application*

This case study will leverage the new approach for architecture selection outlined in Chapter 2. In addition, it will look at applications for stakeholder iteration. While the scenario described in Chapter 2 and Case Study 1 is somewhat idealized due to clear, quantifiable requirements, the practical reality is that stakeholders and decision makers will likely not agree to be committed to the results of an analysis until they actually see those results. For this reason, the results should be considered preliminary until validated by the approving authority. This gives the opportunity for that authority to modify the weightings or the objective function if the final results

are judged inadequate to support the decision. This would also be an opportunity to challenge and adjudicate any driving critical assumptions.

Given the flexibility of the MBSE-enabled optimization, this type of iteration is actually not exceptionally difficult since most of the effort was in setting up the model in the first place. Therefore, the approving authority may be able to go through several iterations of an objective function, with each iteration and weighting being correspondingly rigorously documented. Ultimately, an objective best-value measure will be reached that all stakeholders and the approving authority can agree on.

## **4.2 Case Study 2 Research Setup**

### *4.2.1 Evaluation Method Grading*

The goal of this part of the research is to compare the new approach outlined in Chapter 2 to the traditional AoA approach. Based on original interviews with several experts in architecture evaluation [98], the following set of criteria were developed to compare the approaches:

- Objectivity: do different stakeholders come to the same conclusions when they review the evaluation?
- Repeatability: can the analysis be performed multiple times by different parties and arrive at the same results?
- Transparency: is it clear to any stakeholder who reviews the architecture evaluation why the chosen result was selected?
- Flexibility: does the architecture evaluation easily allow for changes in scope during the evaluation execution?



- Resource Intensiveness: how much does the evaluation cost, how long does it take, and does it require any hard-to-acquire resources (specific expertise, unique IT requirements, etc.)?
- Selects Best Value: Does the evaluation consistently select the best value solution as defined by all the stakeholders?
- Uncertainty Quantification: is uncertainty in the process and results understood so decision makers can have confidence in the results?

In order to compare the two approaches on these criteria, this case study was conducted, both using the new modified approach and the more traditional approach allowing an objective comparison of the results and methodologies.

#### *4.2.2 Overview of Scenario and Reference Architecture*

The case study to compare the methods of evaluation is a representative scenario for a potential U.S. DoD AoA. The scenario in question is notional to avoid sensitive data, but is defined to highlight the applicability and utility of this technique while utilizing publically available information. The technique is scalable to a very thorough modeling application that would be expected of an AoA as demonstrated by [29]. An extension of this dissertation could be pursuing the exploration of these first results to greater details.

The scenario chosen (identified as “IRSat”) is the acquisition of a new Electro-Optical (EO) monitoring capability to support anti-piracy operations in an Area of Responsibility with rapidly escalating threats. The requirements included rapid revisit over an equatorial theater with a comparatively low fidelity sensor to detect watercraft engaged in suspected piracy activity. Given

the heightened attention accorded to this region, standing up a capability quickly has a greater priority than in normal acquisitions.

There were several options in the tradespace to be considered for this AoA. A small constellation of comparatively cheap Low Earth Orbit (LEO) satellites could be built to provide the necessary revisit rate, varying in orbital altitudes and constellation size. As an alternative, a single, large, geosynchronous orbit (GEO) satellite could also provide the coverage. A further option with the GEO satellite would be the opportunity to have an aggregated space vehicle, sharing costs, with a new GEO communications satellite that was planning to enter development at the same time. Capability could also be varied between threshold and objective values. Finally, this scenario incorporates ongoing DoD emphasis on common modular spacecraft parts, which must be considered as part of the scope of the AoA.

Following the methodology outlined in [16], an initial reference architecture was developed in order to establish a baseline for modeling trade study excursions. Utilizing MBSAP terminology [16], a Domains Composition Block Definition Diagram (BDD) is shown in Figure 17 and an activity diagram outlining the basic collection behavior for IRSat is shown in Figure 18.

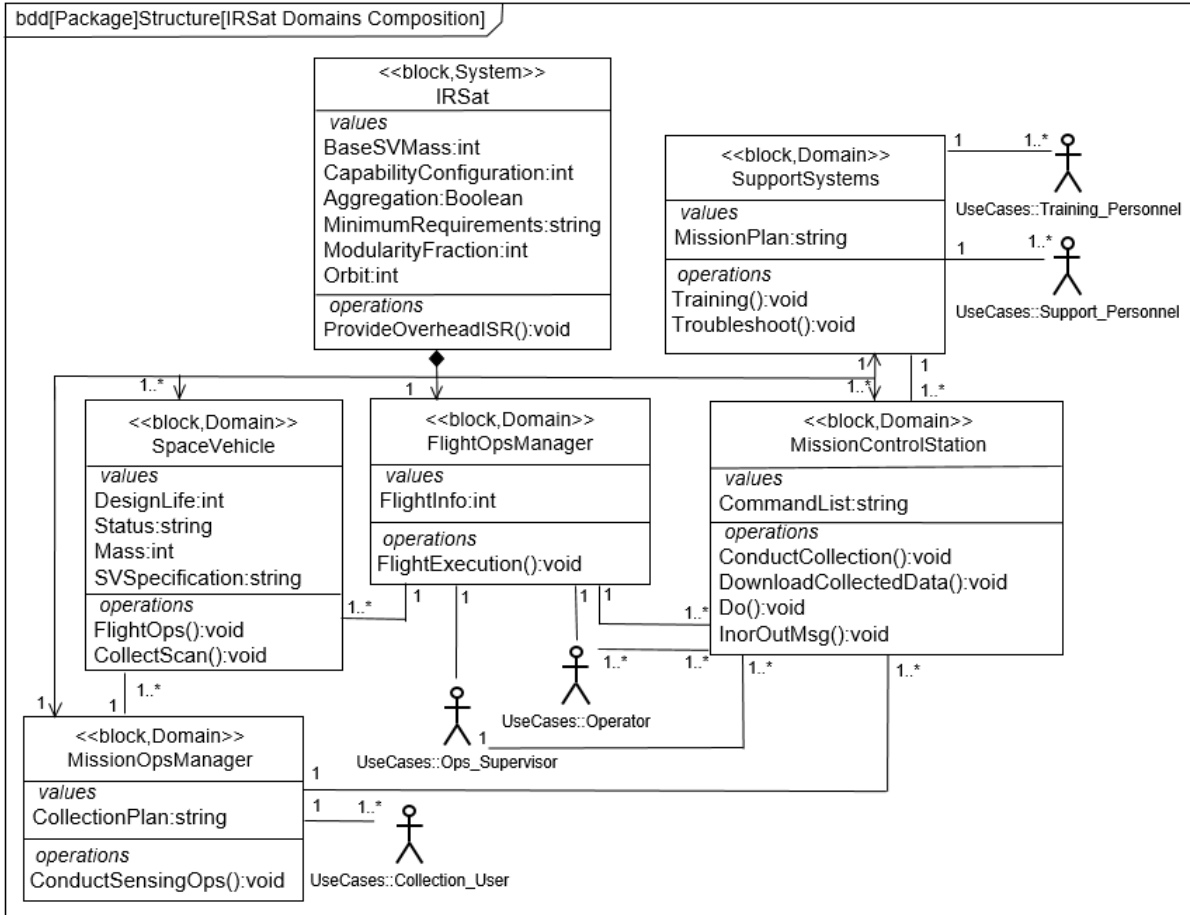
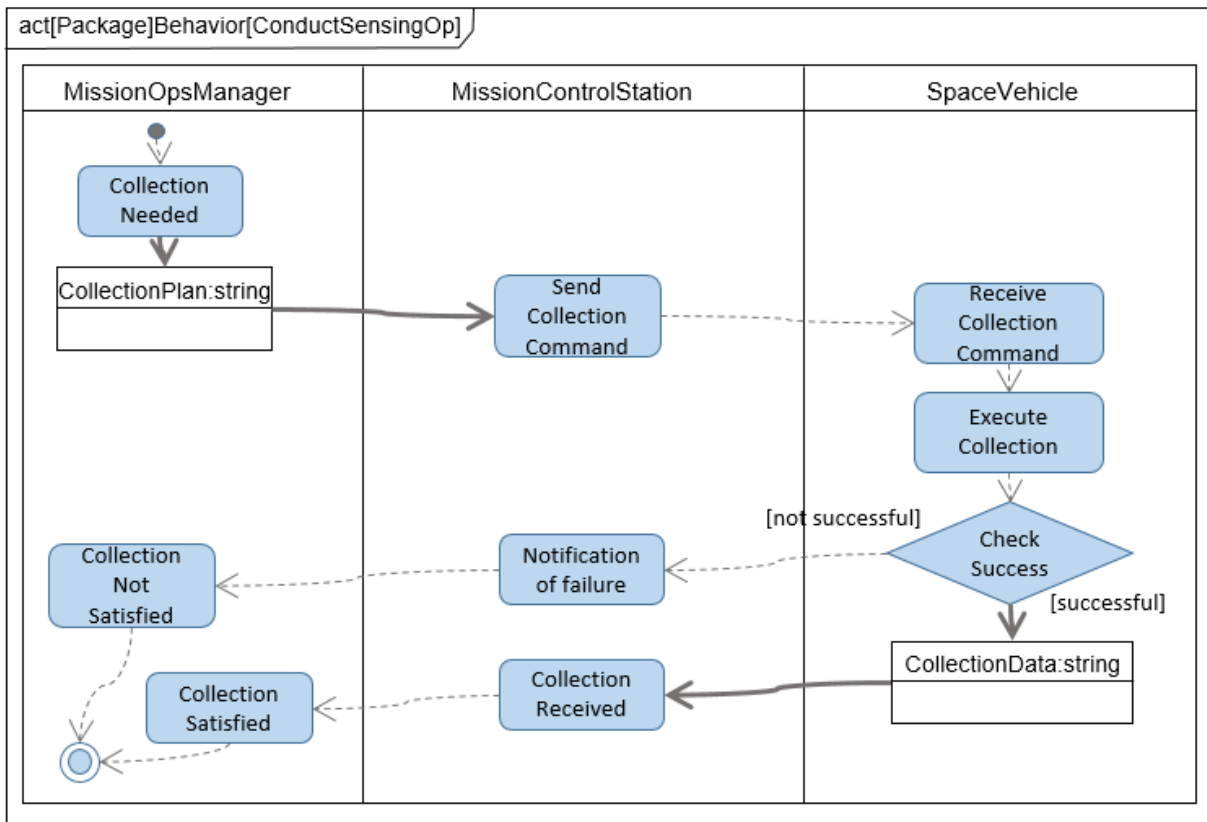


Figure 17: IRSat Domains Composition © [2018] IEEE



**Figure 18: IRSat "ConductSensingOp" Activity Diagram © [2018] IEEE**

To show how the various trade parameters affect the baseline reference architecture, a Variability BDD was developed in accordance with the technique developed in [33]. This is shown in Figure 19, and clearly identifies how varying the 4 input variables trace to select a specific configuration of the reference architecture.

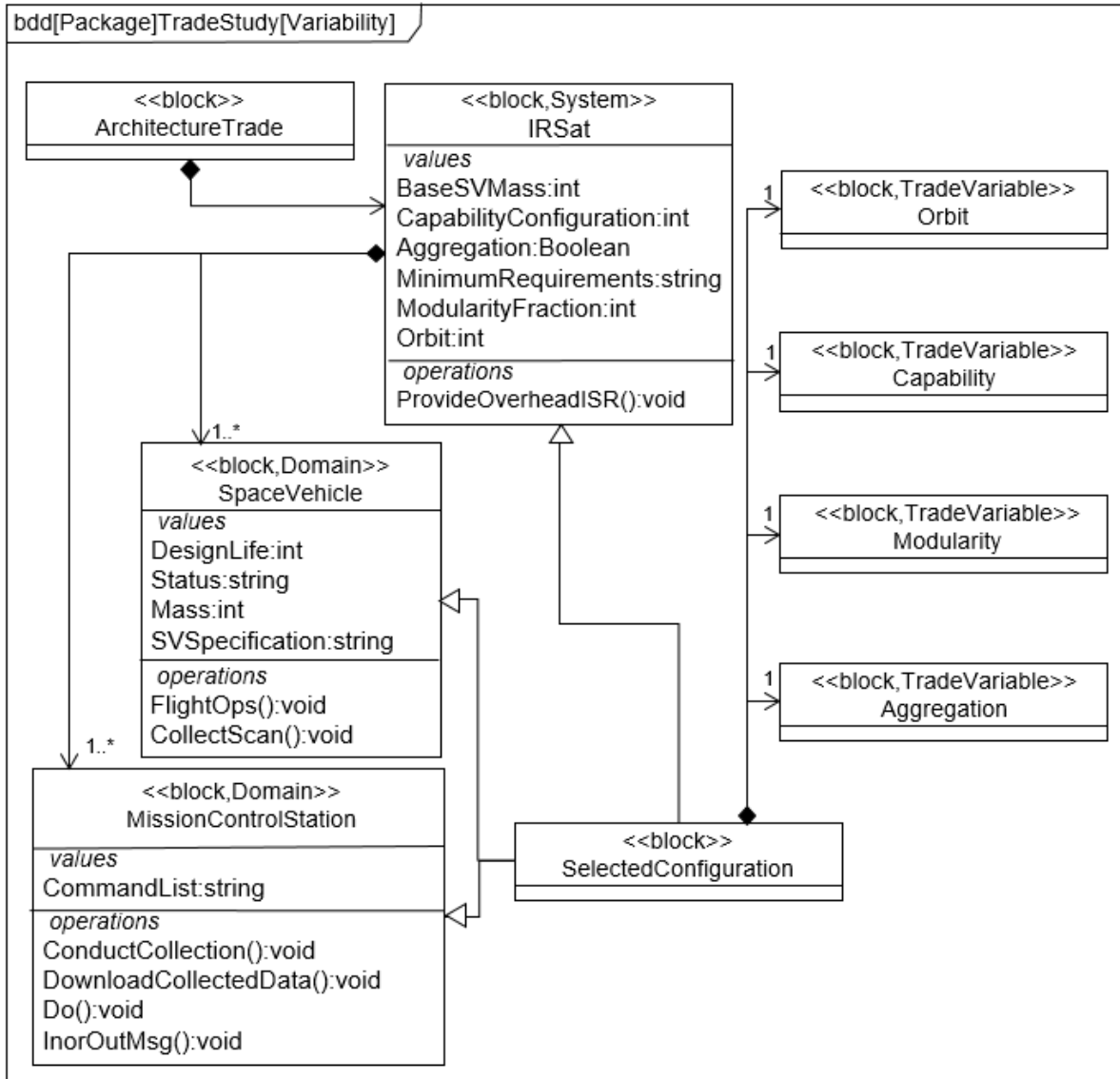


Figure 19: IRSat Variability BDD © [2018] IEEE

#### 4.2.3 Optimization Setup

This scenario led to the identification of several parameters of interest to be included in an AoA. In addition to lifecycle cost, these included capability development schedule, additional capability beyond threshold (minimum capability required), and inclusion of modular spacecraft parts. In a real world AoA, additional trades could be expected, and this methodology is scalable to readily incorporate them.

Under the traditional AoA process, the various materiel solutions would be independently evaluated on the parameters of interest and costed with the results debated until the key decision makers could reach consensus. Under the modified AoA process outlined in section 2, an optimization defining the best value for the Government was selected with associated weighting. Normalization factors were also selected for the contributing analyses represented as functions in this multi-objective optimization. This structure is shown in Equation (9):

$$\text{Eq 9. } \min \left[ \frac{.4}{60} f_1(x) + \frac{.3}{50} f_2(x) + .2 f_3(x) + \frac{.1}{.3} f_4(x) \right]$$

subject to:

$$f_1(x), f_2(x), f_3(x), f_4(x) \geq 0$$

$$f_2(x) \leq 200$$

$$f_3(x) \leq 1$$

In this equation, which is a variant of Equation (2), function 1, with a weighting of 0.4, corresponds to time to first satellite launch from contract authority to proceed and is normalized for 60 months, function 2, with a weighting of 0.3, corresponds to annual architecture cost and is normalized about \$50M, function 3, with a weighting of 0.2, corresponds to minimizing the difference between actual performance and objective (desired) performance with threshold (minimal) performance normalized to 1, and function 4, with a weighting of 0.1, corresponds to a modularity ratio normalized to 0.3. All the functions had a lower constraint of 0, and cost had an upper constraint of \$200M/yr to represent a departmental budget limit. Performance had an upper constraint corresponding to the normalized threshold performance of 1 with 0 corresponding to objective performance.

#### 4.2.4 Contributing Analysis Selection

##### 4.2.4.1 Altitude and Mass Contributing Analyses

The initial mass contributing was developed using surrogate mass fractions and other data from the earth observing FireSat example (for a LEO EO satellite), Defense Support Program (DSP) DoD satellite (for a GEO EO satellite), and the Tracking and Data Relay Satellite (TDRS) (for GEO communications satellite) found in [62]. For the LEO regime, a selection of Walker-orbit constellations ranging from 3 to 9 satellites corresponding with altitudes of 1150 to 660 km as identified in [29] as a realistic set of optimum EO LEO constellations were considered. Walker circular orbit patterns have become popular for systems required to provide continuous or near continuous coverage [99].

##### 4.2.4.2 Cost Contributing Analysis

A number of contributing analyses and supporting models were used in the optimization. The principal spacecraft cost model is the same as shown in Equation (4) in Section 3.2.5.2 from Case Study 1 and was taken from [92].

This is a simplified parametric model based on spacecraft mass ( $m$ ) in kg but certainly a more thorough bottoms up cost estimating methodology could be used as shown in [29] if that level of fidelity is available. This model includes spacecraft launch costs. To realize a per year architecture cost, the per spacecraft cost was multiplied by the number of spacecraft required then divided by the design life in years to reflect the costs of replenishing failed satellites in orbit. LEO satellites had a design life of 5 years and GEO satellites had a design life of 10 years based on the reference architecture. It is assumed any architecture considered could use existing satellite commanding ground infrastructure and services so that wasn't utilized as an analysis discriminator nor used in the cost estimate. The communications space vehicle was costed with the EO space

vehicle if the aggregated GEO option was chosen or costed separately and added to the EO mission if a disaggregated architecture was chosen, so that both cases reflect the net Government cost for the two missions.

#### 4.2.4.3 Schedule Contributing Analysis

The schedule model used is the same as shown in Equation (3) in Section 3.2.5.1 from Case Study 1 and is from a study of Government satellite acquisition schedule information [91].

This model defines the time in months from contract authority to proceed to first satellite launch, and is based on space vehicle mass ( $m$ ), design life ( $DL$ ) in months and number of mission types ( $MT$ ), which affects this scenario if an aggregated EO/Communications satellite is chosen.

#### 4.2.4.4 Modularity Contributing Analysis

A thorough methodology to account for commonality across modular satellite design is shown in [100]. In particular the flexible approach identified in [101] is attractive due to its simplicity, and from that we derived our estimate of the impact of modularity on cost as shown in Equation (10).

$$\text{Eq 10. } A(\$M)=S * [1-\sum (1-y_j)*k_j]$$

In this model,  $A$  is the cost of the modular space vehicle,  $S$  is the cost of a theoretical completely non modular space vehicle,  $y_j$  is the coefficient of cost for modular component  $j$  (a coefficient of 0.08 was assumed which is consistent with spacecraft studies referenced in [100]), and  $k$  is the ratio to the total spacecraft cost for what component  $j$  replaces within the spacecraft.

Increased modularity can have a negative effect on performance as it is assumed common, modular components would not achieve the same performance as exquisite, purpose-built components. This impact would be determined on a case by case basis for a formal AoA. For our



simplified case study, a decrement of 0.1 multiplied by the fraction of the spacecraft that was modular was applied to the spacecraft performance score to account for this.

#### 4.2.5 Optimization Software Implementation

All the models were integrated using ModelCenter. For this analysis, excel spreadsheets were utilized for each of the contributing and supporting analyses and linked together within the ModelCenter interface. The ModelCenter setup for this optimization is shown in Figure 20.

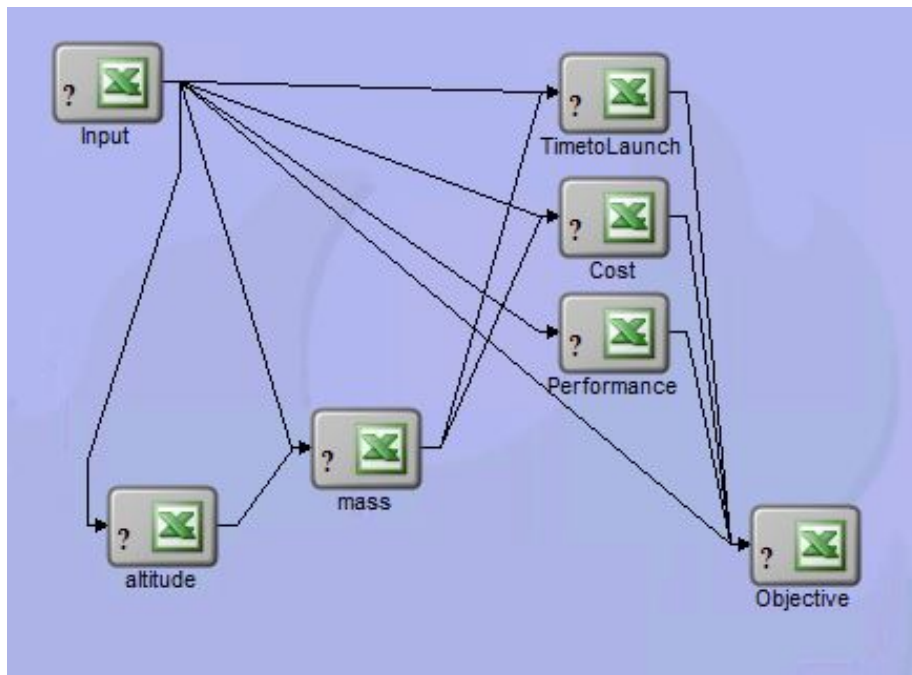


Figure 20: Case Study 2 ModelCenter Simulation Setup © [2018] IEEE

Once a ModelCenter simulation is set up, it can be exercised through the Darwin optimizer and executed repeatedly to ensure consistency and convergence on a global, not local, minimum as discussed in Chapters 2 and 3.

### 4.3 Case Study 2 Results

#### 4.3.1 Simulation Output

For the optimization, after evaluating 877 potential architectures, the best value architecture was identified corresponding to a LEO constellation of 9 satellites with a maximum

modularity fraction of 0.3 and elevated levels of in-built capability capacity to offset the performance decrement due to modularity.

This case study successfully demonstrated the execution of the new methodology examined in this research. Other sources [29] have shown the scalability of this methodology to a wide range of very detailed modeling. As this research progresses beyond this dissertation, further insights and discussion will be developed as more realistic test cases are investigated.

#### 4.3.2 Evaluation Methodology Comparison

The two methodologies were conducted and compared on the evaluation criteria, with the results shown in Table 7. They were directly compared and the methodology that was assessed as concretely better addressing that criteria was marked with a [+], the methodology that performed worse with a [-], and cases where neither methodology conclusively performed better are marked with a [0].

**Table 7: Case Study 2 Methodology Comparison**

Criteria	Traditional AoA Methodology	New Optimization Methodology
Objectivity	[-] Stakeholders may diverge in conclusions based on preferences once the individual analyses are complete	[+] Since all decision makers agreed to the objective function during the buy-in process, they will all have to concur with the final results or agree to iterate the objective
Repeatability	[-] Differing stakeholders may choose to evaluate parameters of interest very differently since qualitative approaches could also be included	[+] While there is still opportunity for divergence, enforcing quantifiable measures reduces the potential for drastic disagreements
Transparency	[-] While the AoA approach and analysis results are documented, how the final decision occurred is often not clear	[+] Methodology approach and results through final decision are documented
Flexibility	[-] Changing a parameter of interest or expanding the input architecture tradespace could often invalidate all work conducted before	[+] As most of the work is up front in setting up the model, that work can be easily reused as the model is modified to accommodate scope changes
Resource Intensiveness	[0] Initially may be easier to set up, but becomes more resource intensive as changes occur to the AoA scope	[0] May require specific MBSE skillsets and IT resources, which should be a negligible cost given the resources of most AoA efforts
Selects Best Value	[-] Typically results in selecting lowest cost for acceptable performance as the	[+] As best value was defined by the stakeholders early in the process, it can be

	quantifiable nature of cost causes it to trump other criteria	justified against the lowest cost acceptable solution
Uncertainty Quantification	[-] Cost uncertainty is typically analyzed and accounted for in a rigorous process, but other measures are not	[+] Mature, quantifiable metrics should have traceable uncertainty bounds similar to the level that cost uncertainty is accounted for

**4.4 Case Study 2 Discussion**

The new optimization methodology was assessed in table 7 performed as well as or better than the traditional methodology in all measures of performance. The following are some highlights.

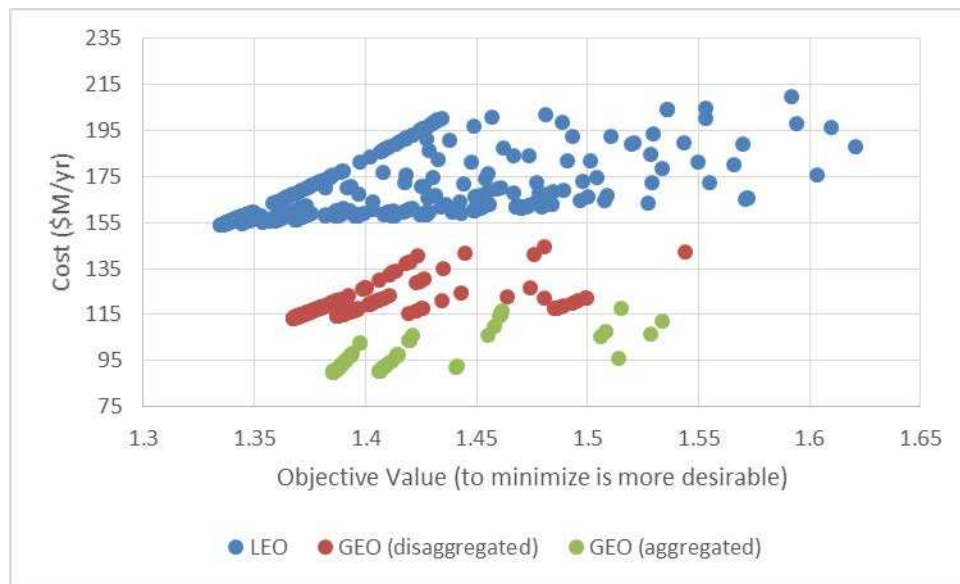
Transparency is frequently mentioned as lacking in the current AoA process, specifically in how it informs the final AoA decision in being reached. There is a sense that in the final decision meetings, which ever stakeholder can make the most convincing argument on the spot for their cause will win the day. It is easy to see how this could be the case given the typical summary chart for a current AoA, which for this case study could look something like Figure 21.

	<b>Lowest LEO</b> (9 Satellites)	<b>Highest LEO</b> (3 Satellites)	<b>GEO</b> (Disagg)	<b>GEO</b> (Aggregated)
Schedule	31 Months	42 Months	72 Months	96 Months
Annual Cost (EO + Comm)	\$155M	\$159M	\$114M	\$90M
Performance	Threshold	Threshold	Threshold	Threshold
Modularity	High	High	High	High
Note: Across all architectures modularity can trade with performance but increases cost as modularity decreases and performance increases				

**Figure 21: Case Study 2 Representative Current Methodology AoA Output © [2018] IEEE**

If Figure 21 was given to a mixed group of stakeholders, it is not clear what the decision should be based on. For example, the perceived value of receiving the capability 3 or more years earlier could be very different across the stakeholders. Since cost is always a high interest topic, in the absence of insights like those produced by the new proposed methodology the lowest cost is usually selected.

In contrast, the new AoA methodology provides a clear identification of best value, rather than a focus only on cost. Figure 22 shows one way this result could be presented to support an architecture selection.



**Figure 22: Case Study 2 Representative New Methodology AoA Output, Cost (\$M) vs. Objective**

Figure 22 is a visualization of all the optimization runs plotted for cost vs. objective score, which is inversely proportional to value as defined by the stakeholders. With this, a stakeholder can understand the interplay between the variables in the trade space, with clear boundaries and logical groupings of options. Notably, this enables the stakeholders and decision makers to actually see how their stated preferences for value as defined when they selected the objective function

compare to cost. Relative interdependencies can be clearly visualized by the slopes of the various curves.

A definitive best choice based on the agreed-upon stakeholder objective is also easily identifiable, which is not the case in the current AoA process. If this identified best choice is not acceptable to the final decision making authority, then this method further provides insight that the stakeholders and decision makers could then use to iterate the weightings and the objective function in order to make an informed modification to the AoA.

The assessment of resource intensiveness was inconclusive, with the limited scope of this analysis, but given the resources involved in a MDAP AoA, in the worst case the new methodology should still be affordable and can reasonably be expected to be considerably cheaper given the significant scope increases in a typical AoA.

A discussion point is that the new methodology will require stakeholder engagement and concurrence earlier in the process. While this may involve increased coordination, it should reduce the amount of discussion at the end of the process as the final, best-value solution will be more readily apparent.

#### **4.5 Case Study 2 Conclusion**

This case study has described a new methodology to conduct AoAs consisting of quantitatively defining best value up front and optimizing for it. On every criteria considered, this new approach performed as well or better than the current AoA methodology which consisted of independently evaluating parameters of interest and cost and attempting to select the best value after the fact. The methodology can be applied to a wide range of other architecture optimization challenges.

## CHAPTER 5: CASE STUDY 3: MISSION CONTROL SOA

This chapter outlines a case study applying the proposed methodology to conduct an architecture evaluation for a software focused problem involving a service oriented architecture in the trade space applied to a satellite mission control segment selection.

### **5.1 Case Study 3 Introduction**

#### *5.1.1 Case Study Focus*

##### 5.1.1.1 Uncertainty Quantification

A major concern for analysis is tracking uncertainty through the proposed methodology. In order to convey confidence in the results to leadership, any proposed architecture evaluation technique needs to be able to demonstrate an understanding of how likely is that the evaluation results will represent the delivered implementation of the final system. While Case Study 2 did investigate uncertainty quantification when comparing the new methodology to the old methodology used in U.S. DoD AoAs, it was simply a relative comparison. Demonstrating that this new methodology better ensures uncertainty quantification by enforcing traceable metrics is the principle objective of this case study.

There are two different types of uncertainty to account for. Type A uncertainty is typically associated with sampling or random error in measurement [102], while Type B uncertainty is related to systematic error and associated with having improperly calibrated measuring tools or using an incorrect model [103]. A robust treatment of uncertainty should account for both types [104]. Several examples of how to approach propagating uncertainty through an analysis are found in Chapters 1 and 2.

### 5.1.1.2 Software

In an effort to demonstrate the flexibility of the methodology proposed in Chapter 2, it was desired that a vastly different scenario be investigated than the tradespace focused on hardware and physical implementation that was demonstrated in the previous case studies. In particular, this scenario is focused on software applications, which is an ever-increasing source of complexity and risk in modern systems. As an example, over 90 percent of the functionality in a 5<sup>th</sup> generation air vehicle such as the F-35 is in software [39]. Given this importance, it would be a critical test of the proposed methodology to demonstrate if software architecting efforts could be adequately handled.

### 5.1.1.3 Cybersecurity

Another major focus area is that of cybersecurity. Cybersecurity is an architectural concern that is continually increasing in importance with recent significant cyber-attacks resulting in multiple billions of dollars in losses and including widespread collateral damage [105]. Cybersecurity is defined as “prevention of damage to, protection of, and restoration of computers, electronic communications systems, electronic communications services, wire communication, and electronic communication, including information contained therein, to ensure its availability, integrity, authentication, confidentiality, and non-repudiation.” [106]

As architectures have grown more interconnected, the cyber “attack surface” or the exposure of the architecture to exploitable vulnerabilities [107], is also forced to grow, making cybersecurity risk management all the more critical in architecture selection decisions. Demonstrating the proposed methodology can handle cybersecurity concerns would be a very relevant and compelling test for this case study.

#### 5.1.1.4 Service Oriented Architectures

A service-oriented architecture (SOA) is the final focus area for this case study. SOA's are an architecting paradigm where functions can be published, discovered, and used as a shared reusable service within the architecture independent of where they are instantiated [108]. They are purported to hold many benefits such as increased flexibility, modularity, and efficiency, and align well with recent trends in both the public and private sectors to reduce stovepipes and capitalize on better information sharing [109]. More variations of SOA designs have been developed to provide services of different flavors, to include Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) [110], which highlights some of the potential of a well-executed SOA approach.

However, while SOA's are a flexible and effective way of operating a networked system, that inherent flexibility comes with many unique security challenges. That same discoverability and portability of services that facilitate their utility can lead to potential cyber attack paths. There are effective methods to mitigate some of these challenges including implementing layered access controls and federated authentication [111]. Nevertheless, the concern of cybersecurity will always be at the forefront given the increased connectivity of modern systems typified by the trend towards service-orientation.

#### 5.1.2 Problem Statement

The scenario chosen for this case study is a notional situation meant to represent a realistic architecture selection involving software, SOAs, and uncertainty quantification in order to meet the desired focus of the case study. While it is not a real-life scenario, it is representative of a realistic situation based on the author's experience and review of appropriate scientific and technical literature [62] [16].



This scenario focuses on an architecture selection decision for the space mission control segment for a scientific research laboratory that controls a number of purpose-built satellite missions. These missions are funded by different customers, and as such have differing requirements for security and acquisition timelines. There is also independent potential for the missions to change requirements or for funding delays and cancellation.

At the start of the scenario there are two missions already in orbit and conducting operations. These missions include an earth observation satellite (“ImageSAT”) that provides high-quality imagery to commercial customers and a scientific research satellite that provides radiation monitoring and analysis of solar flares for a Government customer (“RadSAT”). In addition, a third mission (“TechSAT”) is planned and currently is in the engineering and development phase of system acquisition. This will be an on-orbit technology demonstrator for the laboratory itself, primarily demonstrating proprietary autonomy and on-orbit servicing technology which the laboratory eventually desires to develop into a commercially viable service.

The earth observation mission and the solar radiation monitoring mission both use the research laboratory’s ground infrastructure for communicating with the space vehicles. This is done through separately scheduling time for use of the laboratory’s single Ku-band parabolic antenna dish located on a remote mountain top. While the laboratory operates both spacecraft, the software and mission control stations developed to conduct telemetry, tracking and control functions, as well as mission planning, were developed and are utilized completely independently from each other for both missions. This division was an artifact of the separate contracts, development timelines and divergent security requirements for the two missions.

With the technology demonstrator mission, the customer was internal to the laboratory itself. As an opportunity to realize efficiencies across the ground infrastructure, it was identified

that potentially the new mission could leverage the existing software and mission control stations the laboratory had developed for the other missions. While the laboratory's software engineers were most familiar with developing custom software applications for research, several new hires were very familiar with service-oriented architectures, and highlighted that a SOA approach could make a lot of a sense for this development effort.

However, in addition to the desire to control cost, the laboratory was also very concerned about the cybersecurity considerations for their mission operations. A competing technology firm had recently been the target of a highly publicized cyber hacktivism campaign for selling imagery to the U.S. military. This series of attacks effectively took this firm by surprise and disrupted many of their mission operations, resulting in substantial financial loss from forfeiture of production contracts and reduced customer confidence. This very public example had caused the laboratory to conduct an internal audit which found their ground infrastructure would be susceptible to similar cyberattacks.

Furthermore, the proprietary technology being demonstrated in the mission under development was regarded as critical to the future success of the laboratory and represented a considerable investment of the research and development budget. It could be crippling if this technology was stolen by a competitor or was prevented from being demonstrated in a timely fashion. Given this importance, the laboratory decided that it needed to re-evaluate its mission control infrastructure for cybersecurity considerations for both the old and new missions. A debate arose over whether moving to a SOA helped or hindered this cause. While a SOA may reduce the cyber "attack surface," it was also highlighted that it could increase the consequence of a successful cyberattack as now all the missions shared vulnerability if one was compromised due to the greater interconnectivity through a networked infrastructure. A choice could be made to keep the missions

in separate stovepiped mission control segments to diversify the risk and insulate the missions from each other, which may be more attractive if the missions were of varying importance and would be optimized at different risk postures.

A final consideration for the laboratory involved the flexibility of the mission control segment architecture. Given that the three mission areas supported all had separate customers, their requirements and funding stability were all independently subject to change. It was desired that the architecture selection decision should explicitly account for the ability of the architecture options under consideration to handle changing requirements or funding. This also could support opportunities for technology refreshment, or the ability to integrate completely new missions into the architecture.

In summary, the scenario results in a laboratory decision to conduct an architecture selection for an update to its space mission control segments in preparation for a third space mission coming online. The decision trade space had to consider a SOA approach, and cover the criteria of cost, cybersecurity both from an information assurance and denial of service consideration, and ability to handle changing requirements, funding, and extensibility. This architecture selection evaluation needed to be conducted with a robust treatment of uncertainty in order to convince the laboratory's leadership that the decision was robust in the face of changing circumstances.

## **5.2 Case Study 3 Research Setup**

### *5.2.1 Mission Control Segment Reference Architecture Generation*

The first step of the architecture selection methodology is to define the RA for the problem space under consideration. For this problem, this architecture selection involves defining the

mission control segment for three satellite missions: “ImageSAT,” RadSAT,” and “TechSAT.” A brief description of the three missions follows:

- ImageSAT provides high quality, visible-spectrum imagery to commercial customers on an on-demand basis. Its mission control segment includes unique functionality to plan, schedule, and optimize imagery collections as well as manage and troubleshoot the optical payload.
- RadSAT provides constant monitoring for space weather events to a Government customer. Its mission control segment includes unique functionality to manage and troubleshoot its scientific payloads.
- TechSAT is a technology demonstrator for autonomy and on-orbit servicing technology for the laboratory itself. Its mission control segment includes unique functionality to manage and troubleshoot the prototype payloads and flight software and provide advanced diagnostics of the on-orbit demonstrations.

The current mission control segment (MCS) controls ImageSAT and RadSAT through completely independent command strings, and will have to provide mission control functionality to TechSAT in time for its final ground integration testing prior to launch. The existing MCS utilizes the same antenna, terminal, and modem to communicate with all space vehicles and that is not under consideration to change for the launch of TechSAT. However, the existing MCS software and user terminals are completely separate across ImageSAT and RadSAT and that is under consideration to change with the launch of TechSAT.

Following MBSAP, the first step to generate a RA is to document requirements [16]. Given the scenario for this case study, the applicable RA is for the combined MCS, whether integrated or separate, of the ImageSAT, RadSAT, and TechSAT missions. This RA is responsible for the top-level requirements as shown in Table 8.

**Table 8: Combined Mission Control Segment Requirements**

1) Provide telemetry, tracking, and control functionality for the ImageSAT, RadSAT, and TechSAT missions
2) Provide mission specific functionality such as mission planning, payload troubleshooting, and mission monitoring capability to ImageSAT, RadSAT, and TechSAT
3) Provide Cybersecurity protections to reduce cybersecurity risk to an acceptable level
4) Enable timely and cost effective future architecture changes for technology refresh and to incorporate new missions

In a typical satellite system, the MCS will be responsible for telemetry, tracking, and control (TT&C) functionality, working with the TT&C subsystem on each satellite, as well as any mission specific functionality. TT&C comprises the functions associated with flying the satellite bus, and includes basic status of health information, troubleshooting, and standard bus commands such as those for orbit maintenance. Mission specific functionality can be associated with a number of functions depending on the mission and typically involve controlling the payload. This can include collection management (covering planning, scheduling, and optimizing) for sensor missions, payload status of health and troubleshooting. Mission specific functions can also include unique secondary requirements such as more thorough status of health monitoring if necessary [62].

In order to ensure functionality was comprehensively addressed, a more in-depth review of mission operations functions was researched. In [112], 13 potential functions were identified as

part of mission operations for a space mission. These include nine data processing functions (Mission Planning; Activity Planning and Development; Mission Control; Data Transport and Delivery; Navigation and Orbit Control; Spacecraft (bus) Operations; Payload Operations; Data Processing; and Archiving and Maintaining Mission Database) and four support functions (Systems Engineering, Integration and Test; Computers and Communications Support; Developing and Maintaining Software; and Managing Mission Operations).

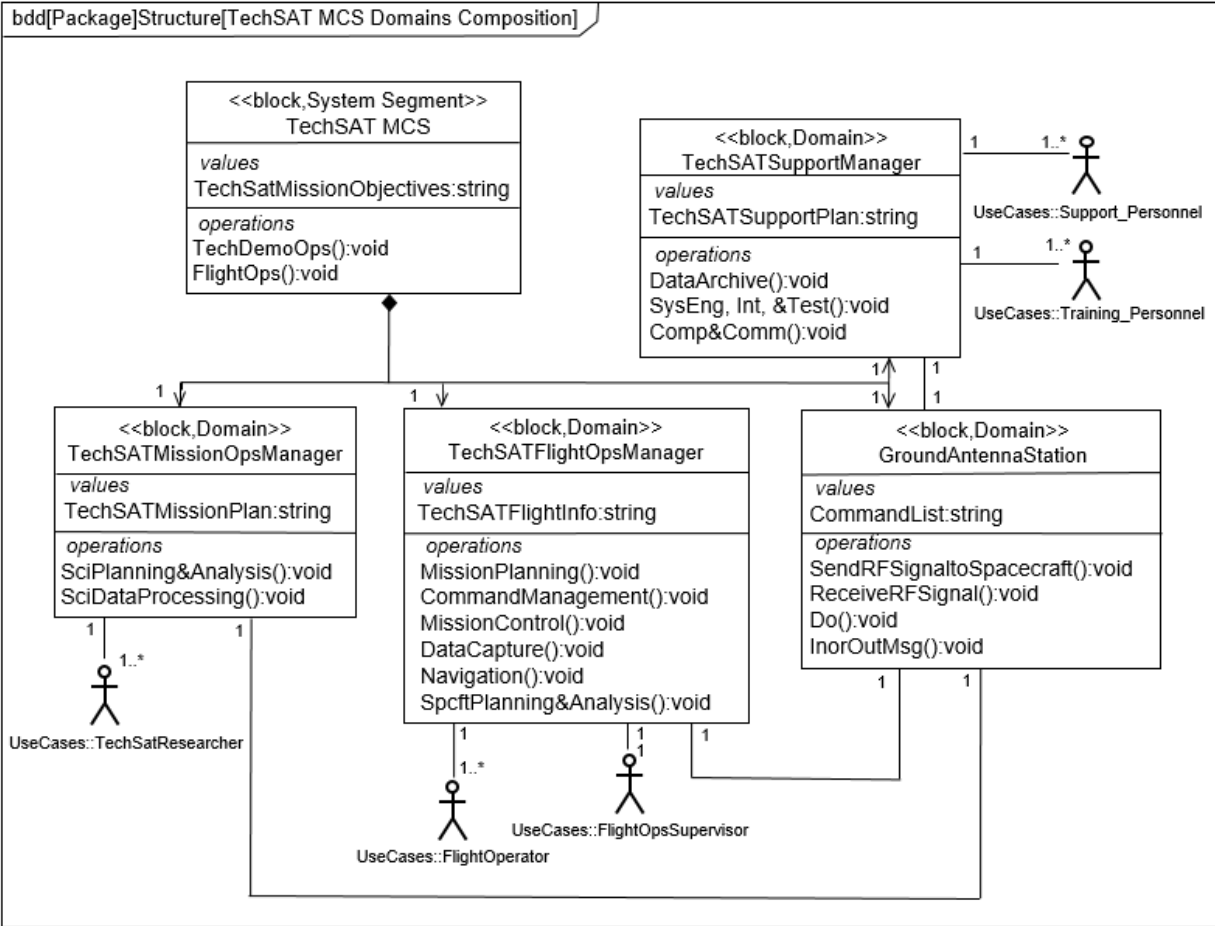
Not all of these identified functions are necessary for every mission, and any mission operations infrastructure should be tailored to the needs of the mission. For instance, a simple academic research satellite could constantly broadcast passively collected data and have no propulsive capability. Such a system would have minimal or no need for a Navigation and Orbit Control Function, or a Mission Planning Function, among others. On the opposite end of the spectrum, some systems may have exceptionally complex functionality in certain areas, which results in much greater complexity and cost for that function than average. As an example, a very complicated and uniquely built scientific collection platform could have a much more costly Payload Operations Function than would be typical for a satellite of comparable size due to unique and complicated custom payload control software.

Despite this tailoring, all of these functions could potentially interact with an enterprise MCS infrastructure solution. It was assumed for this case study that the Managing Mission Operations Function, which is typically a leadership role to maintain the strategic perspective and budget, would be handled through the laboratory's corporate business processes and not through the enterprise MCS infrastructure. Therefore, this function will not be included in the architecture tradespace.

Several of these functions could be allocated in part or completely to the space segment rather than the enterprise MCS infrastructure. For instance, comparatively simple autonomy capability could be used to aid in station keeping for the space vehicles on orbit, which could reduce needed functionality of the Navigation and Orbit Control Function on the ground. These types of trades involve allocating costs between the space segment and ground. Frequently, if disciplined systems engineering processes are not followed, cost savings are pursued during space vehicle development that then lead to unforeseen complexity and large cost impacts during mission operations or system sustainment. [62]

For this case study, it is assumed that the space vehicles are either already launched or otherwise not able to be modified to support modifying the MCS functionality due to existing requirements and stressing technical demands on the space vehicles. While this does limit the trade space of this case study, in the author's experience this is not an unrealistic constraint. This could occur when different acquisition authorities are acquiring the space vehicles and the MCS system. For purposes of the case study, this is assumed to be the result of different business units within the laboratory acquiring the MCS and the TechSAT space vehicle, and ImageSAT and RadSAT are already on orbit and can't be modified, which leads to an inflexible allocation of requirements to the MCS.

A block definition diagram showing the domain composition for an independent TechSAT MCS reference architecture is shown in Figure 23.



**Figure 23: Case Study 3 Independent TechSAT MCS BDD**

Figure 23 shows all the necessary functionality for the TechSAT MCS. As it was observed by the laboratory leadership, a number of the functions, particularly those in the TechSATFlightOpsManager and TechSATSupportManager, were very comparable to corresponding functions in the RadSAT and ImageSAT MCS's. An evaluation of what a SOA MCS would look like resulted in the SOA MCS domains composition shown in Figure 24.



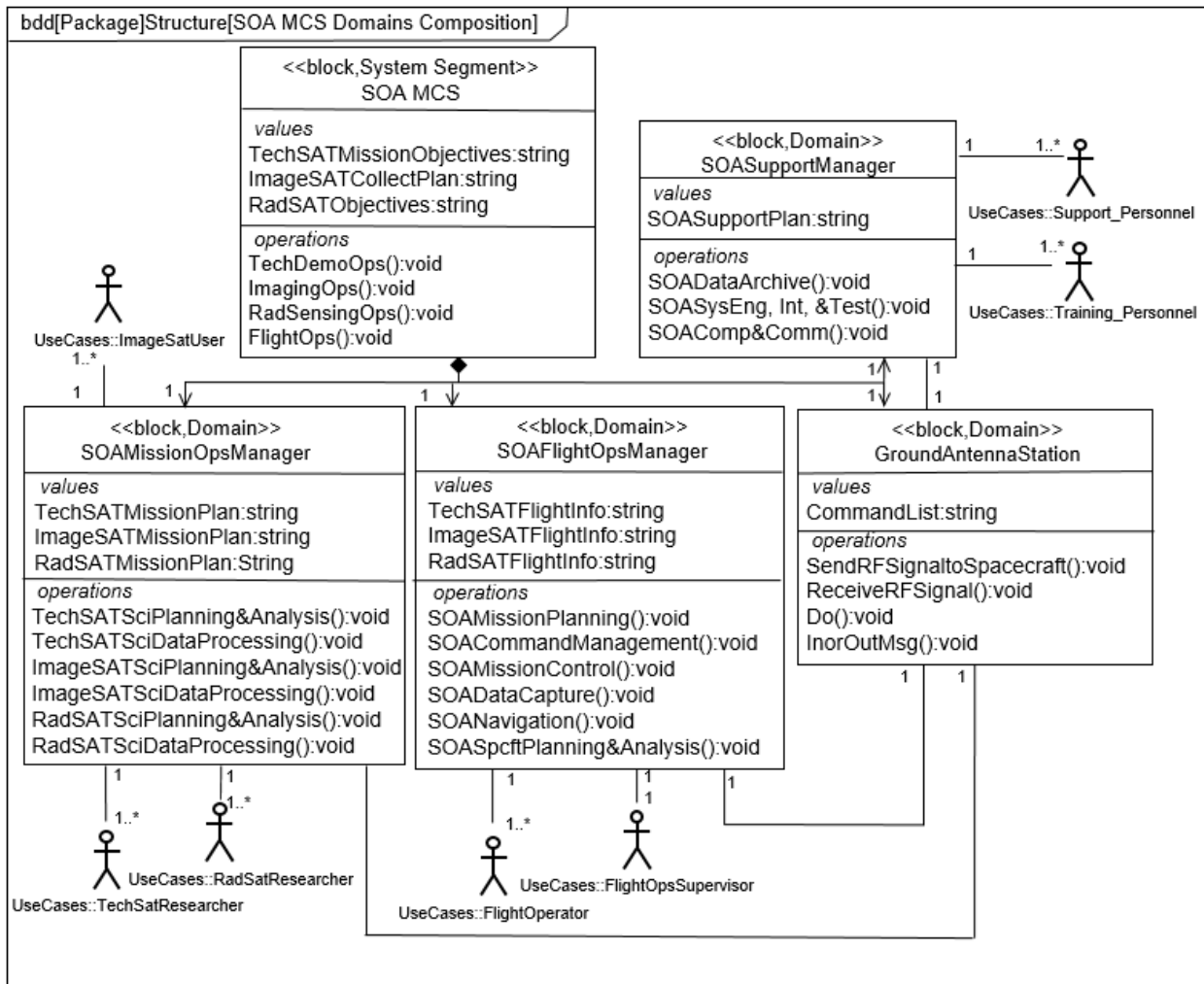
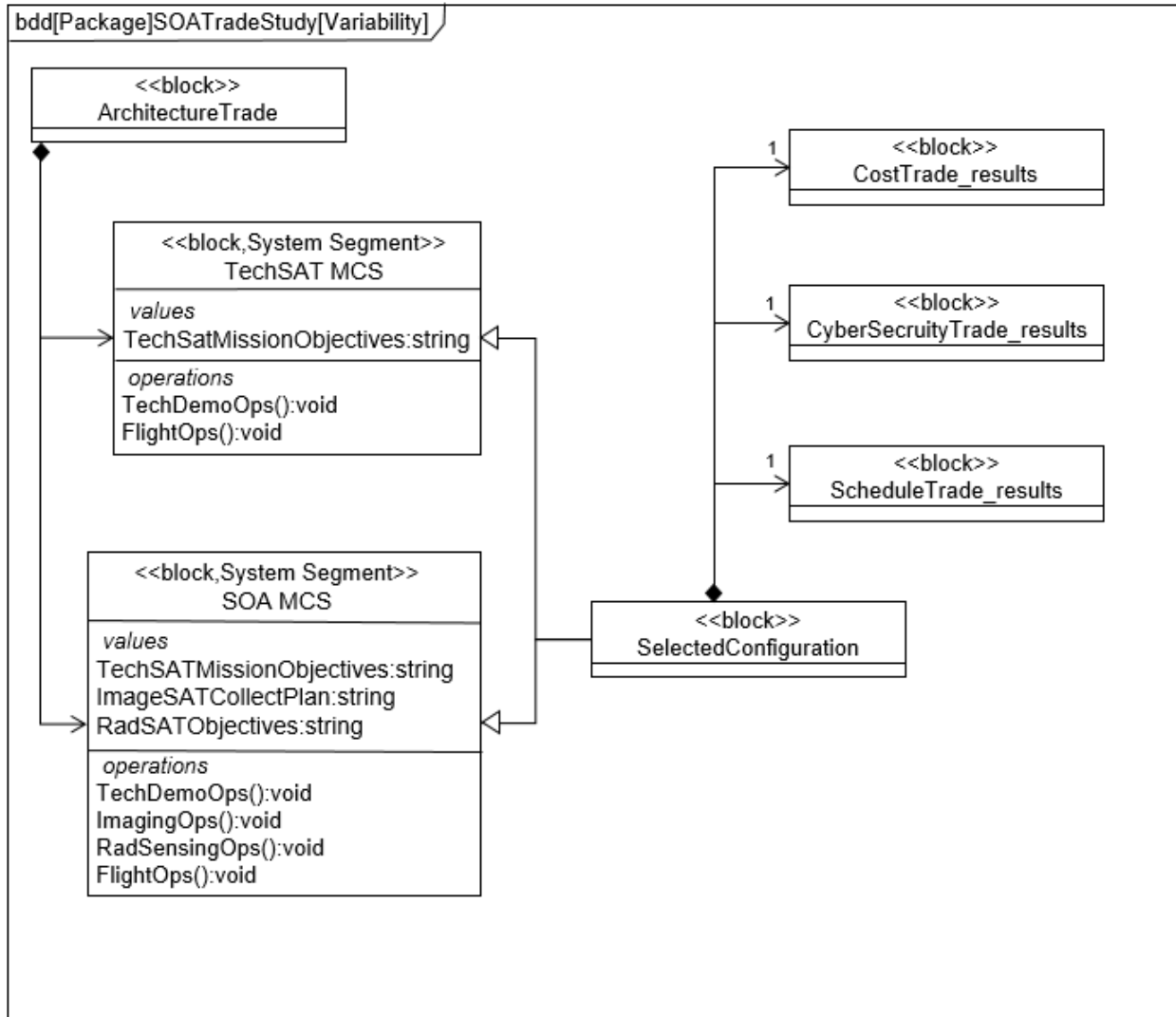


Figure 24: Case Study 3 SOA MCS BDD

The MCS shown in Figure 24 collapses the functionality of the RadSAT, ImageSAT, and TechSAT MCS's into one MCS. This provides an alternative architecture for evaluation in this case study. Following the methodology established in Chapter 2, a variability BDD is defined in order to show traceability between variables and the reference architecture for the architecture evaluation, which is shown in Figure 25.



**Figure 25: Case Study 3 Variability BDD**

This variability BDD is different from that in Case Study 1 (Figure 13) and Case Study 2 (Figure 19) because Case Study 3 is essentially selecting between two substantially different architecture configurations. Each configuration is evaluated on the trades of interest as part of the overall trade study.

### 5.2.2 Optimization Setup

Following the methodology outlined in Chapter 2, an optimization function is defined for this architecture selection. This is constructed in the format shown in Equation 1. This function is

selected based on the needs of the stakeholders, primarily the leadership of the laboratory for this case study. The output of the optimization will correspond with the architecture recommended. Following the example from Case Study 2 in Chapter 4, if this recommended architecture does not meet the stakeholder desires, the objective function can be iterated in a traceable fashion and re-evaluated.

Given the stated objectives and desires of the laboratory leadership, several parameters of interest were selected for inclusion in the objective function. This included annual mission operations cost, cybersecurity risk, and technology insertion schedule. The relative weightings proposed for the initial evaluation of the objective function are given in Table 9. Additionally relevant normalization factors were decided on by stakeholders to ensure the variables were scaled appropriately to each other within the objective function.

**Table 9: Case Study 3 Objective Function Weighting and Normalization Factors for Parameters of Interest**

Parameter of Interest	Weighting	Normalization
Mission Ops Development Cost ( $f_1$ )	0.4	1 / \$2.4B
Cybersecurity Risk ( $f_2$ )	0.3	1 / \$2.4B
Tech Insertion Schedule ( $f_3$ )	0.3	1 / 2 years

The three parameters of interest are defined by contributing analyses in the evaluation of the objective function. The normalization factors addressed what stakeholders established as the baseline measurement for the variables; \$2.4B was the laboratory’s associated mission control segment budget and 2 years was the laboratory’s current average technology insertion timelines to integrate a new mission. In keeping with one of the focuses of this case study, these contributing analyses must also include some way to address uncertainty to enable propagation of uncertainty through the objective function. The functions defining the specific contributing analyses must be

defined first before the final objective can be defined in order to identify appropriate normalization factors for approval by the stakeholders.

This optimization has the potential to be very computationally intensive. As the focus of this research was not on integration of advanced computation techniques and high performance computing resources, this analysis will leverage notional, yet realistic models that require lower computational resources when necessary. However, this technique is certainly extensible to high performance computing as integration of optimizations has been thoroughly demonstrated at these high levels of complexity and resource requirements, including uncertainty analysis. An example of a high-performance computing optimization capability with uncertainty analysis is Sandia National Laboratory's Dakota tool [113].

### *5.2.3 Contributing Analysis Selection*

#### *5.2.3.1 Mission Operations Cost Contributing Analysis*

The first contributing analysis to be investigated is the upfront development cost for the MCS solution of choice. A number of relationships were defined to come up with the cost model for both the stovepiped TechSAT-only MCS solution and the broader SOA solution.

It has been shown that development cost and annual sustainment cost for space missions operations can be realistically predicted as a percentage of the total development cost of the spacecraft [62]. This correlation is due to the relationship that a more expensive spacecraft typically is more complicated and has more mission critical functions and operations that need to be planned for. Also, as the space vehicle represents a greater investment for an owning organization, mission operations have a lower risk tolerance and a correspondingly higher amount of oversight, which drives more costs. [62]

In order to address SOA architecture costs, a Divide and Conquer approach is recommended [114]. In this methodology, a SOA architecture development cost estimate is broken down into component parts which are then independently costed and recompiled into a full architecture cost estimate. For software components, the migration to a SOA framework can fit into four categories as follows [114]:

- Available Service: already existing suitable service from a legacy SOA system or third party
- Migrated Service: a traditional software component that can be used in a SOA with some modification or wrapping
- New Service: a service that needs to be completely developed
- Combined Service: a service that needs built from the combination of any of the above, and could be further deconstructed from the above

For any of these components, a conventional software cost estimation methodology such as the Constructive Cost Model (COCOMO) [115] is suitable to be independently applied with the final results then compiled into the full architecture cost estimate [114]. This means that for the purposes of this analysis, individual service functions will be independently evaluated and compiled into a final cost estimate. Similarly, uncertainty can be evaluated at the system level or independently calculated for the development and/or transition of the individual software components to services and compiled for the final estimate [48]. The migration approach required for each software component will affect the complexity of the development effort and the cost model.

Cost models have been developed that decompose the MCS relative costs by functions [116]. This approach enables a method to evaluate costs for a SOA, as redundant functions can be identified and collapsed in the SOA as the separate missions are integrated. There is also an allotment for various complexities for each function, which can serve as a surrogate for the required migration approach to a SOA. A cost model for development of the TechSAT MCS broken out by functions is shown in Table 10. A separate cost model for developing TechSAT as a SOA suitable for all three missions and transitioning the non-redundant functions in ImageSAT and RadSAT to that SOA is shown in Table 11. Development complexity is assessed based on expert judgment taking into account mission type and SOA transition methodology.

In keeping with the estimate formatting in [62], the final cost estimate for each architecture is given as a percentage of annual operating cost which in turn is based on a percentage of development cost. For this case study, this base annual operating cost was \$5M.

**Table 10: TechSAT MCS Development Cost Functional Breakout**

<b>Function</b>	<b>Development Complexity</b>	<b>Base Cost %</b>
Management	N/A	N/A
Mission Planning	Typical	78
Command Management	Typical	96
Mission Control	Typical	146
Data Capture	Typical	62
Navigation	Typical	78
Spacecraft Planning and Analysis	Typical	63
Science Planning and Analysis	High	662
Science Data Processing	High	480

Data Archive	Typical	18
Systems Engineering, Integration, and Test	Typical	197
Computers and Communications	Typical	7
Total (% of base)		= 1887%
Total (\$M)		= \$94.35M

**Table 11: SOA MCS Development Cost Functional Breakout**

<b>Function</b>	<b>Development Complexity</b>	<b>Base Cost %</b>
Management	N/A	N/A
SOA Mission Planning	High	169
SOA Command Management	High	334
SOA Mission Control	High	410
SOA Data Capture	High	86
SOA Navigation	High	212
SOA Spacecraft Planning and Analysis	High	162
ImageSAT Science Planning and Analysis SOA Transition	Typical	87
ImageSAT Science Data Processing SOA Transition	Typical	181
RadSAT Science Planning and Analysis SOA Transition	Low	0
RadSAT Science Data Processing SOA Transition	Low	0
TechSAT Science Planning and Analysis	High	662
TechSAT Science Data Processing	High	480
SOA Data Archive	High	59
SOA Systems Engineering, Integration, and Test	High	437

SOA Computers and Communications	High	25
Total (% of base)		= 3304%
Total (\$M)		= \$165.2M

Another aspect of the model is uncertainty. While the specific cost model used above did not include uncertainty metrics as described in [62], uncertainty metrics for similar cost modeling strategies are available. For instance, a cost model that breaks mission operations into 91 functions individually assessed for complexity successfully predicted operations cost to within 25% for 13 out of 14 case studies [62] [117]. This can be directly translated into an uncertainty metric and provides a representative surrogate for this case study. This results in a normal distribution for the cost with the mean as the total cost provided in Tables 10 and 11 and a standard deviation of 13.86% of that mean.

#### 5.2.3.2 Cybersecurity Risk Quantification

Cybersecurity is the practice of protecting sensitive information resources from threats. Cybersecurity risk management requires a rigorous and expertly executed ongoing process of risk assessment, governance, safeguards and updates to establish and maintain a secure information environment [118]. The U.S. Department of Commerce National Institute of Standards and Technology (NIST) provides extensive guidance on the conduct of cybersecurity risk assessments [119]. At a high level, a cybersecurity risk assessment should be conducted in alignment with organizational processes and in light of organizational objectives. An information system cybersecurity risk needs to be translated to a mission/business process risk and then to an overarching organizational risk. In this manner, a cybersecurity risk can be interpreted into a quantified organizational impact [119].



An example cybersecurity risk assessment process is typified by the Microsoft Threat Modeling Process (MSTMP) [118]. This is a five step process that includes:

- 1) Identify Security Objectives: what information needs to be protected?
  - 2) Application Overview: how is that information used and instantiated?
  - 3) Decompose applications: identify trust boundaries, data flows, entry and exit points, and external dependencies
  - 4) Identify Threats: what applicable threats exist against your applications?
  - 5) Identify Vulnerabilities: map threats to current architecture to establish which threats and potential countermeasures are most warranted for consideration
- Once these steps are completed, the final step is to close the loop back to Application Overview to identify design changes and evaluate whether implemented countermeasures are sufficient

A critical piece for the proposed methodology is being able to translate cybersecurity risks into a quantifiable metrics with associated uncertainties. Fortunately such techniques to perform a cybersecurity risk assessment to result in a quantified measure of risk have been developed. In fact, insurers have made these types of assessments in order to develop cyber insurance products [120]. For instance, an example that is in alignment with the NIST guidance and MSTMP-type processes broke cybersecurity attacks against financial institutions into six categories (virus attacks, denial-of-service attacks, financial fraud, system penetration, theft of proprietary information, and unauthorized access), then used Computer Security Institute-Federal Bureau of Investigation (CSI-FBI) data to create validated generalized linear models to quantify the cost

impacts associated with the potential attacks. This was then used to evaluate the feasibility of cybersecurity insurance as a means of Risk Transference [121]. It has been highlighted that a cybersecurity quantification metrics can be integrated into a SOA [122].

Cybersecurity risk models can be decomposed based on confidentiality, integrity, or availability (commonly known as “CIA”) as well as other measures. Any decomposition needs to be based on clear, observable, and useful measures [123] and on aspects where there exists knowledge since otherwise it is not of much benefit [52].

Once the cybersecurity posture of a system is decomposed to a sufficient level of fidelity the next step is to quantify those risks. At a top level, a quantified cybersecurity risk has two components; a likelihood of occurring over a period of time and an impact which is typically translated into dollars lost. Both of these should have some sort of distribution associated with them since neither is known with absolute certainty. Establishing appropriate distributions for these two aspects is a critical step in this case study. [52]

An appropriate method to assess likelihood of a cybersecurity breach is to use data from a relevant set of similar situations. There is a significant amount of historical information available to pull from such as metrics on data breaches within various industries from the Verizon Data Breach Investigative Report (DBIR) which not only has metrics on number of breaches in industrial categories but includes information on the type of breach as well [124]. A suitable way generate a distribution in a conservative fashion is to apply Bayesian logic to create a Beta distribution for the likelihood of occurrence. This is done by taken an “uninformed prior assumption” (manifested as a Beta distribution with  $\alpha$  and  $\beta$  both set to 1 which results in a uniform distribution from 0 to 1) as a conservative starting point for uncertainty and updating with relevant metrics (add number of successful cybersecurity breaches to the  $\alpha$  and the remainder of the sample

size to the  $\beta$ ). This will result in a Beta distribution that captures the likelihood of the cybersecurity risk occurring within the time period of interest. [52]

In order to quantify the impact, subject matter experts (SMEs) can provide an appropriate distribution with an uncertainty bounds. Research shows experts can be calibrated to provide this estimate for a risk with an appropriate confidence interval. While it may be a large confidence interval, that is not in of itself an issue as long as it can be quantified, and there are often rational ways to quantify the range for the distribution. For instance, a data breach distribution could have a minimum impact associated with writing up an incident report to a maximum impact associated with performing some sort of public affairs “penance project” in order to recapture public trust (research so far does not seem to indicate cybersecurity breaches have a major long term impact on stock prices, any such impacts appear transitory [52]). In this way a confidence interval can be established with bounds. A common distribution then to be fitted to this confidence interval is the lognormal distribution in order to capture high-impact events. [52] It should be noted that a national security scenario would likely be analyzed differently due to the threat and national security implications, but this scenario focused on a commercial laboratory.

Calibrated SME judgment can also be used to establish the likelihood distribution if relevant historical information is not available. Also, events can be used to update and provide a more accurate or precise Beta distribution. A common example would be a “red team” penetration test, the success or failure of which could update the Beta distribution in accordance with SME judgment [52].

For this case study risks were broken down and assessed in two broad loss categories: denial of service and industrial espionage. These were assessed for four large software groups: each stovepiped mission, and the SOA as a joint software project for a total of 8 cybersecurity

risks each with a likelihood and impact distribution that were then folded into the overall optimization.

In this case study, it was assumed that information on a population of similar scientific research institutions or software products was available to provide representative sample, even if it was small. This resulted in the distributions described in Tables 12 and 13 below:

**Table 12: Stovepiped Cybersecurity Risk Likelihood Metrics**

Risk	Successes	Population Size	Beta dist. $\alpha$	Beta dist. $\beta$
ImageSAT Denial of Service	5	24	6	20
ImageSAT Business Espionage	2	6	3	5
RadSAT Denial of Service	41	52	42	12
RadSAT Business Espionage	2	5	3	4
TechSAT Denial of Service	0	5	1	6
TechSAT Business Espionage	0	6	1	6

**Table 13: SOA Cybersecurity Risk Likelihood Metrics**

Risk	Successes	Population Size	Beta dist. $\alpha$	Beta dist. $\beta$
SOA Denial of Service	0	5	1	6
SOA Business Espionage	0	6	1	7

Cybersecurity risk impact distributions were created by having a calibrated expert estimate the risks for each threat with a confidence interval. This was then converted into a lognormal distribution in order to adequately capture risks that can have large architecture impacts in keeping with modeling cybersecurity [52]. This distribution was then truncated at a max result of \$2.4B to ensure the highest impact to which was assessed to be the upper loss the company could experience

(total bankruptcy). This ensures an unrealistic result (a loss far in exceedance of anything at stake) did not skew the average return. The confidence intervals used to calculate the lognormal distributions associated with the risk impact distributions are shown in Tables 14 and 15.

**Table 14: Cybersecurity Risk Impacts for Stovepiped Architecture**

Risk	90% CI Lower Bound	90% CI Upper Bound
ImageSAT Denial of Service	\$500	\$2,000,000
ImageSAT Business Espionage	\$500,000	\$600,000,000
RadSAT Denial of Service	\$500	\$45,000,000
RadSAT Business Espionage	\$500	\$5000
TechSAT Denial of Service	\$500	\$3000000
TechSAT Business Espionage	\$500,000	\$750,000,000

**Table 15: Cybersecurity Risk Impacts for SOA**

Risk	90% CI Lower Bound	90 CI Upper Bound
SOA Denial of Service	\$500	\$9,000,000
SOA Business Espionage	\$500,000	\$1,350,000,000

The bounds provided in Tables 14 and 15 came from SME judgment as follows. The minimal loss from a cybersecurity incident was at least \$500 which was associated with filing an incident report. TechSAT and SOA Business Espionage cases had a higher minimum associated with having to conduct a new business case analysis with trade secrets being loss (even if that information was never acted on). The upper bound was based on the expert-assessed criticality of timely service and the sensitivity of the information. ImageSAT had a higher Business Espionage upper bound due to the sensitivity of some of the customers of the ImageSAT system. RadSAT had a higher Denial of Service impact due to the risk that the entire mission could be invalidated

if too much data was not able to be collected. TechSAT had a higher Business Espionage impact due to the sensitivity of the technology involved. The combined upper bound SOA impacts were summations of the corresponding impacts associated with all of the constituent systems.

It should be noted that this is a representative top down model at the system level meant to capture the differences between the two architectures, and it is expected that in a real world cybersecurity risk assessment a greater level of fidelity and decomposition could be achieved. This technique is fully extensible to more discrete cybersecurity risks at the component level as shown in [125]. In fact, the Monte Carlo technique for calculating the objective could be applied either directly to the discrete risks, or to a surrogate model of Monte Carlo results for discrete risks [104].

#### 5.2.3.3 Technology Insertion Model

A representative technology insertion schedule model was developed based on which functions were already provided in the SOA architecture and a comparative level of effort to recreate software functions for a new stovepiped system. A comparison of the functional breakout shown in Tables 10 and 11 was conducted, which showed a mean reduction in cost of 40% to build a MCS for a new mission if it leveraged functionality already provided by the SOA. This was interpreted to directly translate into a reduced technology insertion schedule due to reduced complexity and lower level of effort for the same laboratory software development workforce. A similar normal distribution with a standard deviation of 14% was also used. This is intended to be a representative schedule model but any appropriate schedule model with a distribution could be used.

### 5.3 Case Study 3 Results

#### 5.3.1 Simulation Output

The Monte Carlo simulation was run for 10,000 trials with the results shown in Figures 26 and 27 below.

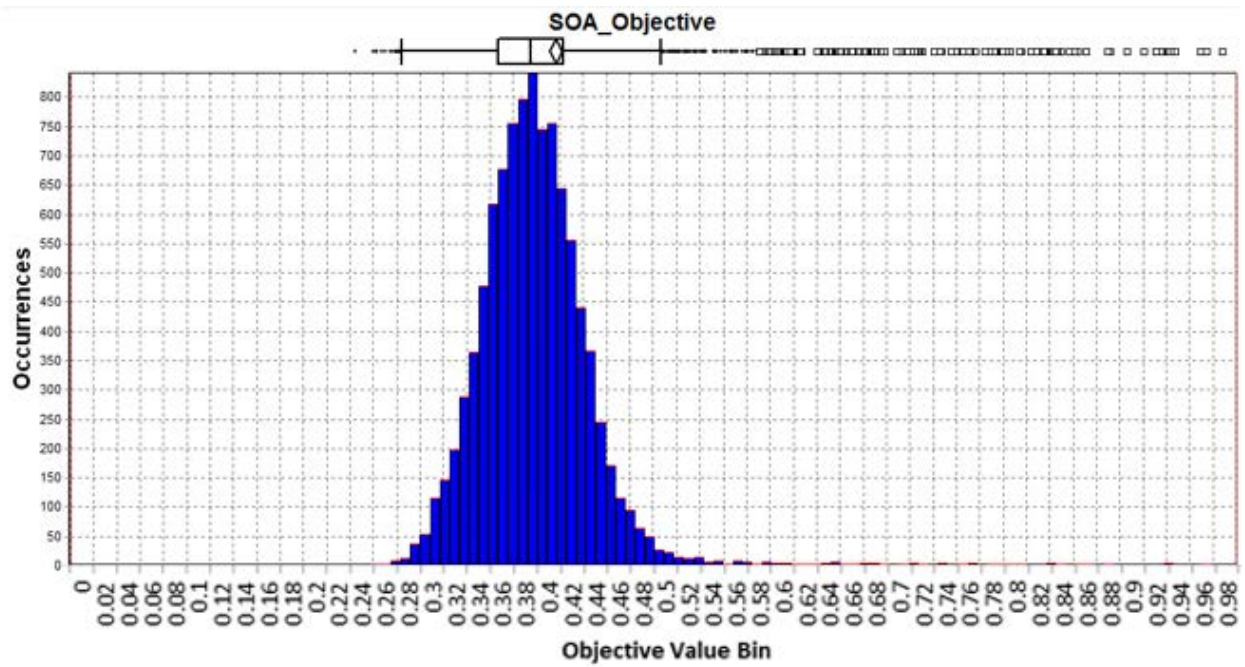
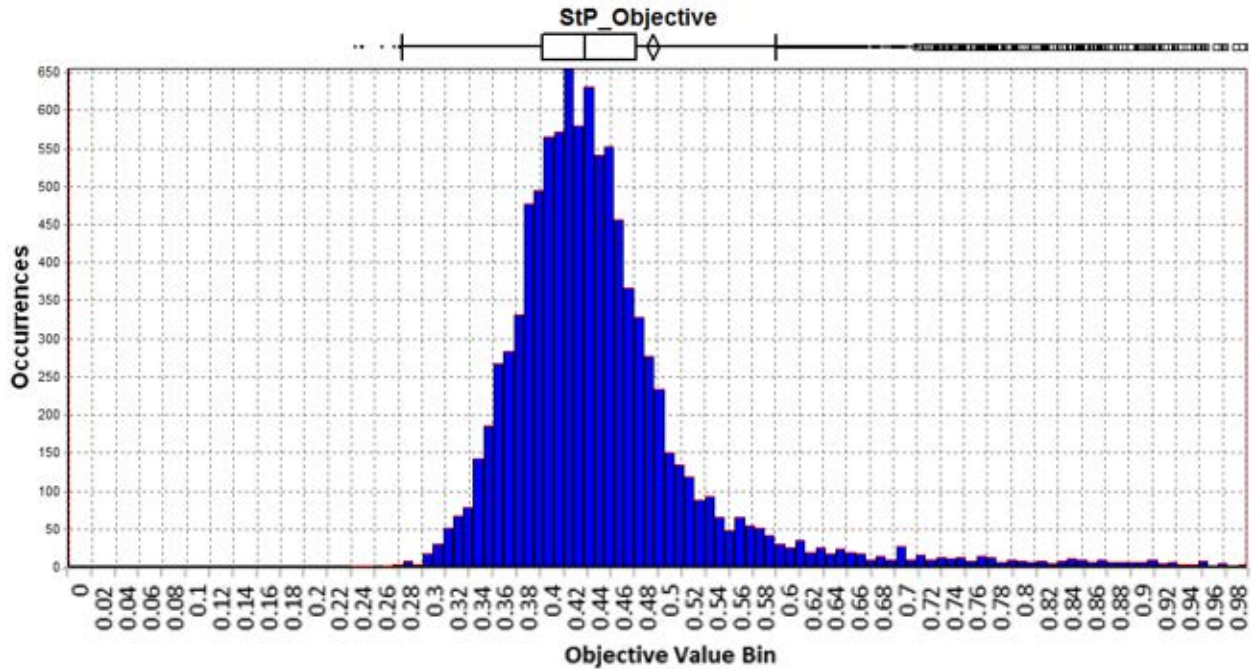


Figure 26: Case Study 3 Histogram of SOA Objective Results



**Figure 27: Case Study 3 Histogram of Stovepiped Architecture Objective Results**

The distributions captured in the above histograms had the parameters shown in Table 16.

**Table 16: Stovepiped vs. SOA Objective Value Metrics**

Architecture	Stovepiped	SOA
Mean	0.50	0.42
Standard Deviation	0.28	0.19
Median	0.44	0.39
Min	0.24	0.24
Max	2.80	2.75

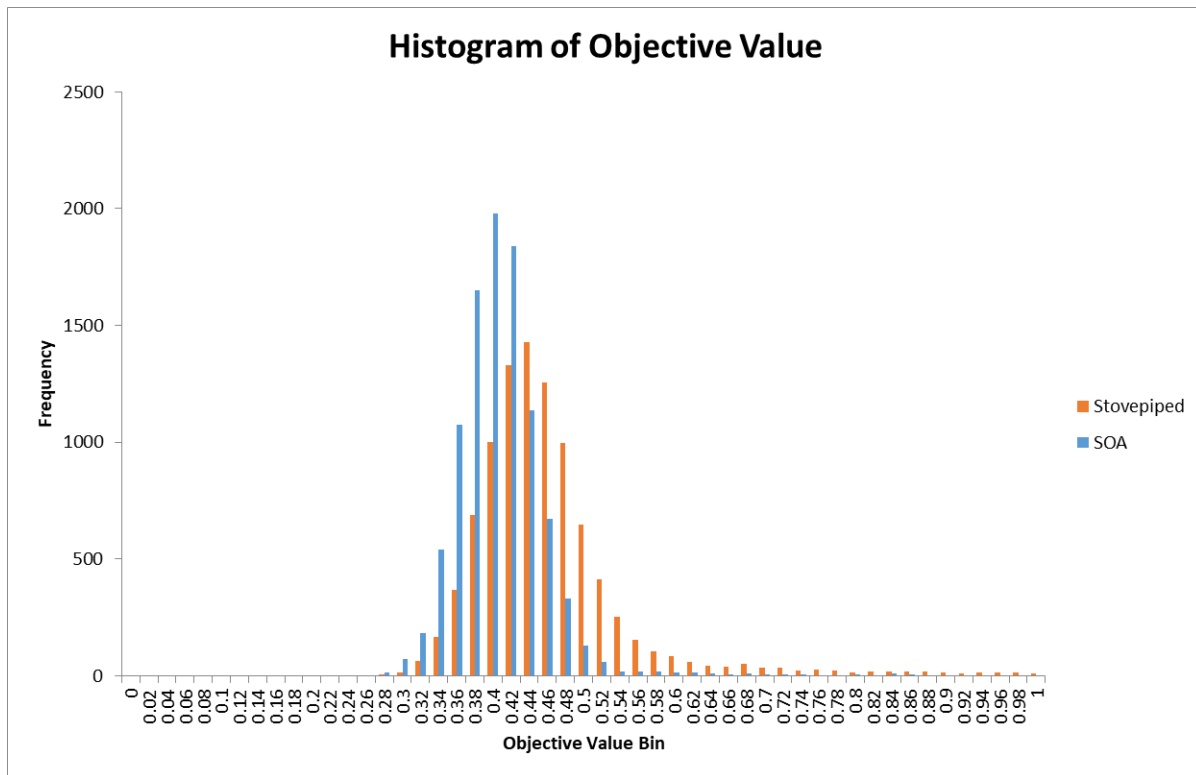
### 5.3.3 Comparison

The SOA architecture had a distribution for the objective that was lower (more optimum), had a lower standard deviation, and a lower median. Choosing the SOA architecture was therefore



not only a better decision for the criteria selected on average, it was also a more robust selection.

A histogram of both plots is shown in Figure 28:



**Figure 28: Case Study 3 Objective Histogram Comparison**

After reviewing the results, it appears that the stovepiped architecture, while it was cheaper to initially develop, retained a lot of legacy cybersecurity risk in the ImageSAT and RadSAT systems that allowed for high-impact cybersecurity breaches at a non-negligible likelihood of occurring. The SOA architecture, which was more expensive initially to develop, directly impacted that legacy cybersecurity risk by updating those systems, greatly reducing the likelihood of those high-impact events. When combined with the benefit from the shorter technology insertion schedule, this was enough to ensure the SOA achieved a better objective value on average and with less variability than the stovepiped architecture.

## 5.4 Case Study 3 Discussion

### 5.4.1 Architecture Selection

Based on the results provided in section 5.3, it appears clear that the SOA architecture is the better choice for the stakeholders as defined in the objective they provided through all the associated metrics of the objective. It was clearly more likely at lower (more desirable) objective values and less likely at high objective values than the stovepiped architecture. However, that only tells part of the story.

Upon reviewing the results, it was somewhat fortuitous that this example had such a clear differentiation, and in fact it is very possible in an architecture evaluation like this for an option to have a lower mean but a higher standard deviation and potentially be more likely at some more desirable objective evaluations. Effectively it could result in a wider histogram bin distribution that is more likely at both the best cases and worst cases. This means that stakeholders should carefully think through what metric associated with the outcome is the one they want to correspond with their selected architecture. It could be the architecture with the better mean, the better standard deviation, the better median, or one selected by some other metric. However, it is important to note that this methodology captures and clearly displays the uncertainty distribution for the objective allowing stakeholders to iterate their selection criteria in a traceable fashion until they arrive at an acceptable result.

### 5.4.2 Satisfaction of Case Study Focus Areas

This case study adequately addressed all four of the focus areas identified. Software assessments are a mature area of study and were directly translatable into the new proposed methodology. SOA's were also directly translatable, and, in fact, the proposed methodology handled SOA's well by enforcing a functional decomposition through MBSE, enabling the clear

tracking of functions across stovepiped and SOA options which in turn make allocation of contributions to metrics of interest straightforward.

Cybersecurity was also satisfactorily incorporated through the use of techniques to quantify cybersecurity risk based on historical information and expert judgment. This is likely a significant finding, as all too often cybersecurity is not considered in architecture trade studies. In the author's experience, it is just assumed that cybersecurity will be addressed through the application of the Risk Management Framework (RMF) [126] or some other community-standard cyber security management program to whichever architecture is chosen. This means the long-term cybersecurity implications from the architecture decision itself aren't accounted for. It is interesting that cost, which is frequently a driving decision factor for architecture selections, is similarly highly variable and is deemed sufficiently addressed by expert evaluation [64]. Given that, it is likely that cybersecurity risk, if properly quantified, can become more of a major driver in future architecture decisions.

Addressing uncertainty was the main expansion of the proposed technique explored in this case study. In general, by enforcing traceable metrics the new methodology handles uncertainty better than conventional architecture evaluations and selections, which tend to avoid analyzing metrics that have high degrees of uncertainty. This is typified by how cybersecurity, a metric typically considered highly variable, is often viewed as a Boolean variable in most DoD architecture KPP evaluations (the architecture is either compliant or not with Information Assurance guidance) [71]. This unfortunately oversimplifies cybersecurity which is increasingly important in development activities.

### *5.4.3 Utility in Assessment Flexibility and Excursions*

One of the principal purported benefits of this methodology was handling flexibility in the architecture trade studies as identified in Case Study 2, due predominantly to most of the work being contained in the creation of the model setup rather than the execution of the analysis. This allows for simple changes to the model to encompass a wide range of flexible scenarios with relatively little manpower required. Case Study 3 results supported this and it would require comparatively less manpower than a traditionally-conducted architecture evaluation to update the model and execute the analysis again.

Furthermore for Case Study 3, the inclusion of explicitly quantifying uncertainty does enable a wider range of potential further excursion analyses for the architecture decision makers to consider. In particular, sensitivities can be captured that convey how much the analytical elements of the architecture evaluation contribute to the resultant objective value uncertainty distributions that are used as input to the final stakeholder decision. This could allow decision makers, who may be dissatisfied with the level of uncertainty in the results, to fund further analysis, testing, or other research to reduce uncertainty in those high-payoff parameters. This would be especially critical if there were substantial overlap in the objective uncertainty distributions, leading to an unclear decision recommendation.

Through the further application of Bayesian logic, the impact of uncertainty reducing measures can be quantified and propagated through the analysis to see if they sufficiently reduce the end result decision in order to warrant being pursued. The combining of probabilities to update a prior estimate is known as Bayesian Inference, and can be measured by Bayes Theorem shown in Equation (11).

$$\text{Eq 11. } P(A|B) = P(B|A) * P(A) / P(B)$$

Equation 11 calculates how the estimate of the probability of event  $A$  occurring given event  $B$  had occurred [127]. This could also be extended to continuous probability distributions such as those used in this case study as shown in Equation (12):

$$\text{Eq 12. } f_{X|Y=y}(x) = f_{Y|X=x}(y) * f(x) / f(y)$$

In Equation (12), Bayes Theorem applies where  $A = \{X = x\}$  and  $B = \{Y = y\}$ . Since this a probability density, the terms become 0 when the variables are evaluated at discrete values. It is assessed that this can be handled for purposes of a simulation like Case Study 3 by using the functions identified as distribution inputs into the Monte Carlo simulation, which in turn would generate an updated distribution.

As an example extension of the scenario in Case Study 3, assume that the laboratory decision makers were dissatisfied with the amount of uncertainty and overlap in distributions for the objective values in the options evaluated. Upon review of the analysis, it was identified that a significant amount of the uncertainty in the final objective was due to the uncertainty in the cybersecurity risk assessment inputs. It was then identified that there was an option to “buy down” that uncertainty by pursuing a cybersecurity “red team” assessment on prototype versions of an MCS implementation for both the SOA and TechSAT stovepiped configurations, the completion of which with no vulnerabilities identified would indicate the proposed implementations should be at a reduced cybersecurity risk. The laboratory decision makers needed to decide if that increase in fidelity and discrimination in the architecture evaluation would be worth funding the red team assessment.

Using Equation (12),  $f_{X|Y=y}(x)$  would correspond with the updated cybersecurity risk distribution assuming a “passed” red team evaluation,  $f_{Y|X=x}(y)$  would correspond with the distribution associated with cybersecurity risk for systems that pass this type of red team evaluation

(either provided by a calibrated SME or some other source such as historical information provided by the red team),  $f(x)$  would correspond with the original or “prior” uncertainty cybersecurity risk distribution, and  $f(y)$  corresponds with the chance of passing the red team evaluation (again provided by a SME or other relevant data). Notably there could be other formulations for Bayesian Inference that could be used if different parts of the relationship were better understood, effectively swapping elements of what is the input and output of the calculation to better leverage the knowledge available of the situation [52].

For purposes of this case study it is assumed that values for  $f_{Y|X=x}(y)$  and  $f(y)$  could be represented as beta distributions calculated from information of suitable representative populations of previous red team certifications updating an uniformed prior assumption similar to the process for establishing the original value of  $f(x)$ . It was assessed that the same values would apply to all the cybersecurity risks in this case due to the similarities in the TechSAT and SOA cybersecurity approaches, but it could be calculated individually. The alpha and beta values for these distributions is shown in Table 17 below.

**Table 17: Beta Distribution Parameters for Red Team Certification**

Distribution	Successes	Population Size	Beta $\alpha$	Beta $\beta$
$f_{Y X=x}(y)$	1	30	2	30
$f(y)$	8	10	9	3

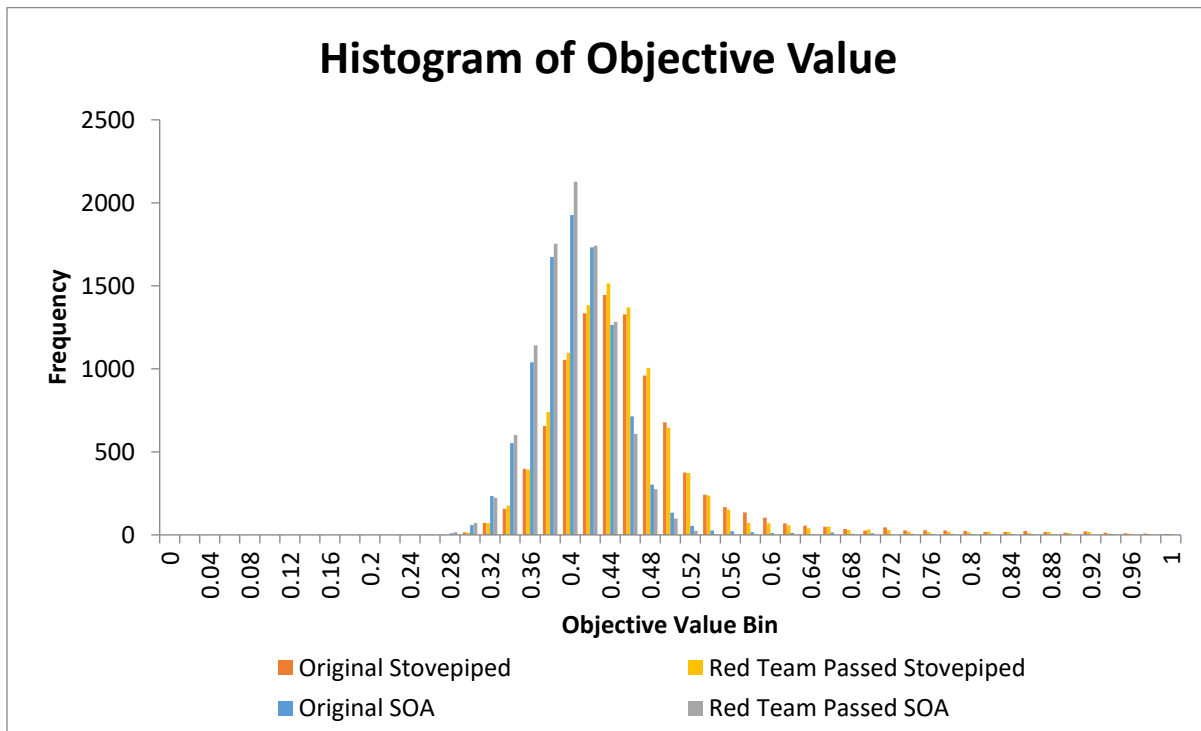
After calculating the new distribution, and leveraging the flexibility of the modeling approach, then the updated cybersecurity risk posture for the TechSAT MCS and SOA MCS can simply be inserted into the simulation and the Monte Carlo evaluation executed again to see how the final results were impacted. This was done using realistic representative distributions for the

elements of Equation (12) for 10000 trials with the results shown in Table 18 which can be compared to Table 16:

**Table 18: Stovepiped vs. SOA Objective Value Metrics for Passed Red Team Assessment**

Architecture	Stovepiped	SOA
Mean	0.48	0.39
Standard Deviation	0.25	0.06
Median	0.43	0.39
Min	0.23	0.27
Max	2.83	2.70

A graphical depiction of the new uncertainty distributions compared to the originals is shown in Figure 29.



**Figure 29: Comparison of Red Team Passed and Original Architectures Objective Values**

Directly comparing the results in Table 17 with the results in Table 16, the discrimination improves between the means (the original results had a difference of 0.0789 compared to the new difference of 0.0862) which corresponds with an approximately 9% improvement in discrimination if comparing by mean alone. However, by far the biggest difference is the standard deviation for the SOA architecture objective value which was reduced by 67% (the Stovepiped architecture objective standard deviation was also reduced, but not nearly as much). So, the effective “overlap” of the uncertainty distributions in the final result is likely reduced by significantly more than just comparison by the mean alone would indicate. Visual inspection of the histograms confirms an improvement in discrimination, although it is not intuitively obvious to the author that it is significant. Upon reviewing the analysis, it is likely the reduction in high-impact cyber-events in the SOA architecture that passed a red team assessment (represented in the long tail on the right side of the histogram) is the principal driver for the difference in standard deviation, which is logical given the nature of the red team assessment.

Whether the effect on the objective uncertainty distributions and improved decision discrimination is worth the cost of the red team assessment would depend on the laboratory management’s desires and the cost of the assessment, but at least leveraging the uncertainty quantification in this methodology would enable a decision informed by the impact on uncertainty the red team would provide. It would be a very worthwhile avenue for future research to investigate how uncertainty distributions should be compared in architecture selections, including how discrimination should be measured and improved. For this case study, since the inputs were directly tied to programmatic measures (cost, cybersecurity cost risk, and schedule), they could be translated to dollar values which would imply the overlap of the final objective distributions could also be quantified as such, enabling a direct comparison to the cost on executing uncertainty “buy-



down” options such as the red team assessment in this example. This would enable a fairly direct and informed decision on whether or not to spend resources on the option to buy down uncertainty.

Notably accounting for the corollary of the uncertainty buy down option (a red team assessment that did discover a vulnerability in this example) should also be accounted for in the value proposition. This would likely require a separate Monte Carlo simulation since otherwise blending the two results would result in a wider uncertainty distribution and imply less insight into the decision, which would not follow Bayesian logic as both cases, discovered vulnerabilities or not, improve knowledge of the decision, reducing uncertainty.

#### *5.4.4 Optimization and Uncertainty Integration*

While Case Study 3 has focused on the selection between two architecture options, the promise of the methodology proposed in this research was to select the optimum architecture based on stakeholder needs, which could include continuous distributions for input variables. Therefore a meaningful extension of Case Study 3 to be investigated would be an optimization for a continuous input variable and the selection of an optimum that is based on some measure of an uncertainty distribution.

A plausible and useful excursion then would be to investigate a trade between permanently expanding the software development team to reduce the technology insertion timeline, which was one of the contributing analyses to the objective. The cost impact of attempting to accelerate software development by leveraging additional manpower has been studied [128] [129] [130], and generally found to be inefficient from a direct cost perspective as more manpower results in greater rework from less shared knowledge. However this inefficiency may be worthwhile in a highly schedule driven field such as technology development and provided that the additional manpower is not added late in the project, which tends to increase project duration [130].

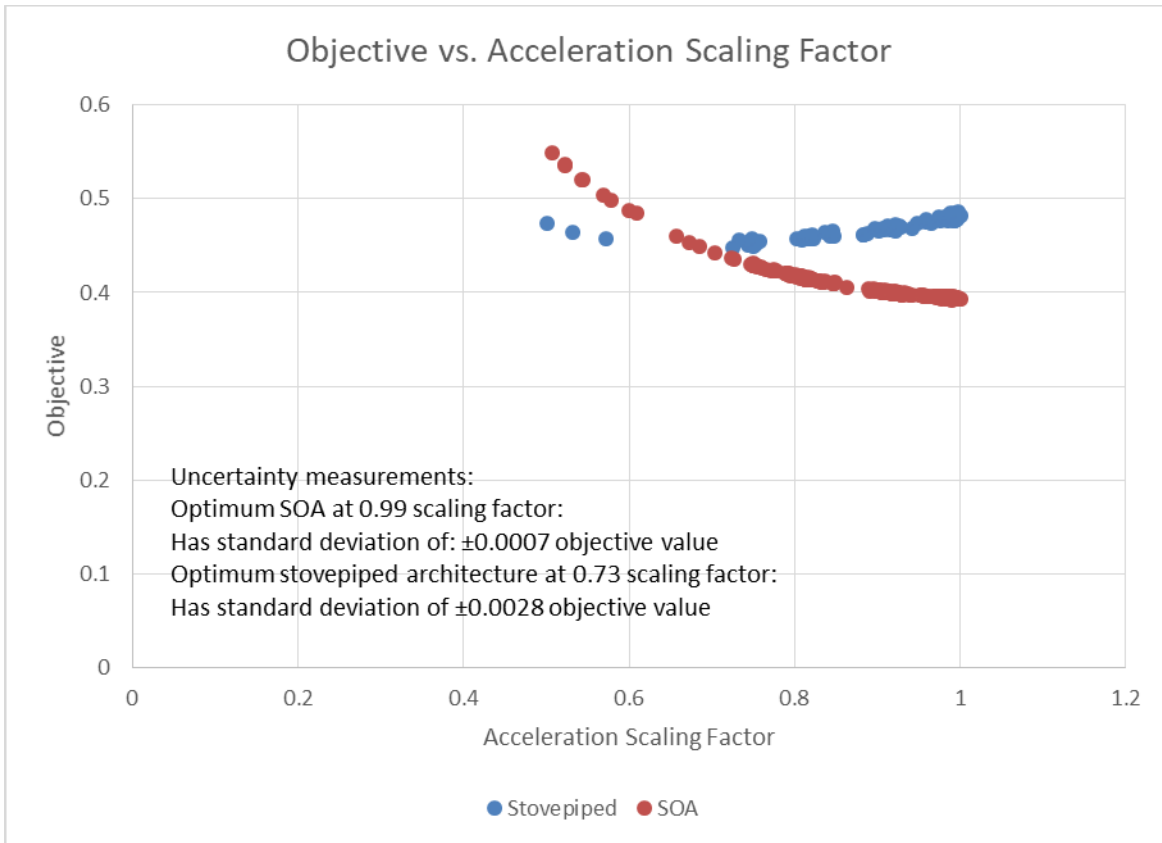
By leveraging the COCOMO II cost model [128], a relationship between schedule reduction through expanded manpower and cost ratios was defined as seen in Equation (13):

$$\text{Eq 13. } y = x^{-1.138}$$

where  $y$  is the cost multiplier and  $x$  represents the acceleration scaling factor (equal to the accelerated schedule duration divided by the baseline duration). As the schedule is reduced due to increased manpower, cost increases through a power function. Essentially this relationship provides another variable for the laboratory management in this case study to control, and while it is a programmatic variable (expanding the standing manpower pool) rather than a technical aspect of the final design, it may alter which technical configuration is the most attractive based on the defined objective function.

This relationship between cost and schedule was inserted into the simulation along with an “acceleration scaling factor” variable to be exercised from 0.5 to 1 along with a Boolean variable of selecting a SOA or not. The optimization was set up similar to Case Studies 1 and 2 with the Darwin algorithm again used as the optimization algorithm. However, for this case study a nested Monte Carlo simulation was run for each configuration according to the input distributions defined in section 5.2.3 in order to account for uncertainty. The mean of 10,000 runs for each configuration was taken as the assessed objective value that would be input into the optimizer for that architecture configuration. A sample size of 10,000 runs was selected as a large enough sample size to achieve a suitable stability in the mean output for the computation resources at hand.

When the simulation was executed, it evaluated 379 scenarios through the full 10,000 Monte Carlo simulations. The results are shown in Figure 30.



**Figure 30: Objective vs. Acceleration Scaling Factor**

In the results, the optimum configuration corresponded to a SOA architecture with a manpower that results in an acceleration scaling factor of 0.9901 which resulted in an objective evaluation of 0.392. Figure 30 uses multiple colors to discriminate the results of the SOA and Stovepiped architecture for ease of viewing, but the two configurations were evaluated simultaneously in the same optimization. Of note, while the SOA outperformed the Stovepiped architecture at the optimum, that wasn't the case across the entire range of possible acceleration scaling factors. This highlights the differing sensitivities the configurations had to changes to the inputs and in their contributing analyses outputs.

The results are interesting for a number of reasons. Clear parabolic trends emerged for both the SOA and the Stovepiped configurations, which would indicate that the number of runs in the nested Monte Carlo simulations likely resulted in means of sufficient stability to conduct a

meaningful analysis and provide visibility to stakeholders. Standard deviation was calculated for this optimum selection with the result being 0.0007. The standard deviation near the optimum solution for most optimum stovepiped architecture result was also low, at 0.0028. However, indications of the impact this underlying uncertainty due to the input distributions caused are evident, foremost being that visual inspection would appear to indicate that a SOA with an acceleration scaling factor of 1 (that is, no acceleration), would be the optimum, however that was not what was actually selected. A SOA with an acceleration scaling factor of 1 was evaluated with an objective value of 0.393. This is very close to the selected optimum and is likely due to the variability introduced by using an input distribution rather than a discrete input.

With this technique, it would appear that the optimization algorithm will always struggle to identify the true optimum once the differences between evaluated objectives falls within the variability in the output introduced by the input distributions. If more computational power is available, more nested Monte Carlo runs could be executed to reduce variability in the means being evaluated by the optimizers. A proposed alternative to this technique could also be to fit a surrogate model to the results and directly solve that for the objective. How to properly set up such a simulation in an appropriate fashion to account for uncertainty and how to interpret the results should be an area for further research.

### **5.5 Case Study 3 Conclusion**

This case study provided an example of how to incorporate uncertainty analysis into the new proposed architecture selection methodology. This new methodology proved very capable of incorporating uncertainty, including, specifically uncertainty of what has traditionally been associated as measurements that are too subjective to be part of an architecture trade study such as cybersecurity. Overall, this case study lends credence to the benefits of this new methodology.

## CHAPTER 6: SUMMARY

This chapter summarizes the results of this research, draws conclusions, and provides recommendations for future research.

### **6.1 Synthesis of Results**

This work explored a new methodology for conducting architecture evaluation and selection activities that integrated MBSE while enforcing selection of quantitative decision criteria by a decision authority in order to mathematically select the optimum architecture. It uses MBSE to frame the flow of the optimization analysis. Since an architecture selection decision occurs very early in an acquisition program, before any system design work has been accomplished, the starting point is a reference architecture that defines the characteristics of the system category in which the acquisition is being executed. A key element of the RA is a high level SysML BDD (see Figure 1) that defines the basic structure of such a system. This is then transformed into a Variability BDD that explicitly declares the specific system configuration corresponding to an alternative under evaluation. Finally, the values from this diagram are input into Model Center, using the Darwin algorithm to compute the overall optimization value for that alternative, which also enables a robust treatment of uncertainty. In all three of the case studies conducted, this was able to select an architecture in a fashion that ensured traceability of the final decision to the original decision criteria.

#### *6.1.1 Case Study Summaries*

Case Study 1 evaluated a satellite communications SoS architecture in light of U.S. Air Force Space Command's SEV initiative. The purpose of this case study was to demonstrate the initial feasibility of this methodology technique to a real world problem. The results of the case

study selected an architecture that relied on a large, expensive space vehicle, which at face value would not achieve the desired strategic direction of the SEV. This reflects the reality and the realized difficulty of achieving the SEV with existing acquisition approaches to satellite constellations.

Case Study 2 evaluated a satellite remote sensing architecture highlighting trades in mission aggregation and modularity. The purpose of this case study was to directly compare the new proposed methodology to the existing method by which architecture trade studies for major government acquisitions are typically conducted. Criteria for this comparison were objectivity, repeatability, transparency, flexibility, resource intensiveness, selects best value, and uncertainty quantification. In all criteria, the new methodology was evaluated to be better than the current methodology with the exception of resource intensiveness which was inconclusive.

Case Study 3 evaluated a satellite MCS SOA decision with a focus towards cybersecurity and uncertainty quantification. The purpose of the case study was to expand the types of architectures evaluated, including criteria that are traditionally thought to be very subjective, as well as to show how a rigorous treatment of uncertainty could be incorporated. Overall this was successful in selecting an architecture while meaningfully communicating uncertainty in the architecture options to the decision authority.

### *6.1.2 Preliminary Validation*

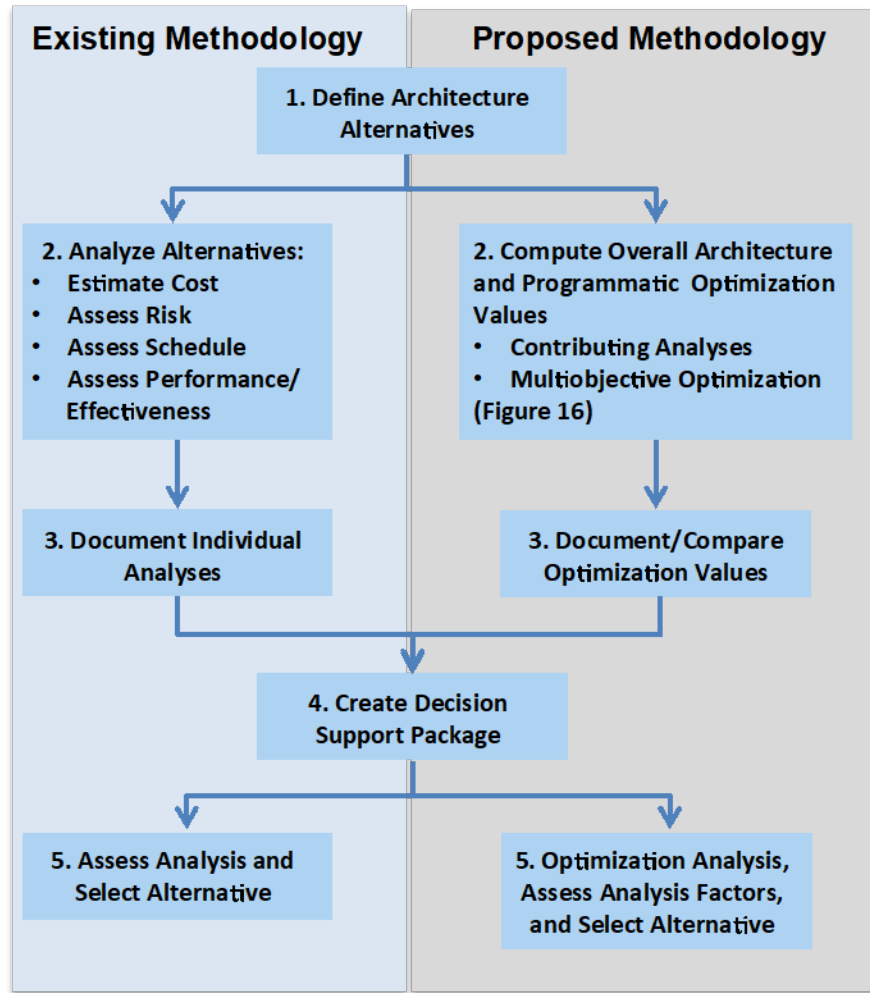
This work has been reviewed and published in several peer-reviewed publications. Some of the case study results are also in line with real world experience which has been highlighted where applicable. Lastly, a panel of five U.S. DoD acquisition experts from Government service or employed by a Federally Funded Research and Development Corporation, who all had experience in architecture selection decisions, was consulted to evaluate this methodology. These

experts all had held senior technical or managerial positions in the aerospace industry, had a minimum of 25 years of experience and three held a Ph.D. in an engineering discipline. While there were some reservations primarily due to skepticism regarding the feasibility of simplifying an architecture evaluation to a quantifiable problem, that view was in the minority and the methodology was assessed to be a useful decision aid.

## **6.2 Conclusions Derived**

### *6.2.1 Enhancement and Comparison to Prior Methods*

The primary contribution of this research lies in improved support to critical decisions in the early stages of an acquisition program arising from an integrated optimization analysis vs. individual analyses of decision factors. Figure 31 summarizes this comparison.



**Figure 31: Comparison of Existing to Proposed Methodology**

On the left, a high level summary of the existing AoA methodology, extracted from the DoD Analysis of Alternatives Handbook [97], is centered on a series of individual analyses of decision factors that include cost, risk performance, and schedule, which are presented to the decision authority. That authority then weighs the analyses and makes a judgment about the preferred alternative. In the proposed approach, an integrated optimization process draws on the best available supporting data and relationships and computes an overall value for each alternative, with factor weights agreed to by program stakeholders. The decision authority can then make a



selection based on a synthesis of the decision factors, and will also have insight into their relative importance. The result is better, more rigorous information support to these critical decisions.

While no methodology is perfect, the proposed approach demonstrably performed better than the current methodology in every comparison criteria with the exception of resource intensiveness which was inconclusive. This criteria is likely less of an issue for major architecture decisions and the new methodology is also likely actually less resource intensive for large architecture evaluations due to scope changes.

Ironically, the main initial criticism received of this methodology is the thought that it oversimplifies a complex problem. However, the current architecture selection methodologies tend to oversimplify even more in practice. When complex parameters such as cybersecurity are even considered in current architecture selections, they are typically included in a very ad hoc, subjective fashion. This is likely not useful since subjective methods of evaluating risk that rely on ordinal descriptions, such as risk cubes, have recently been questioned in utility, with some claiming they actually provide no or negative value due to human tendencies to interpret them differently [131].

In fact simple scoring methods have gained in popularity mostly because they are simple and easy to use, but in actuality appear to have no documented or provable benefit [132]. Paradoxically more mathematical methods may be discouraged because they are more prone to audit when simpler (and much worse) ordinal scales are not. [52]

### **6.3 Recommendations for Future Work**

The methodology proposed in this research demonstrates great promise in improving the quality of architecture selection decisions. This research established the basic methodology, evaluated it through three case studies, and conducted several excursions comparing it to the

existing common methodology and highlighting its utility. However, there is much further work that could be done to confirm and expand on this research.

Basic areas for future research could include applying this methodology to different scenarios and utilizing different tools. More robust sensitivity analysis could be explored, in particular on the weightings and normalization factors selected in the objective function, as that is one of the principal inputs provided by the decision makers. Furthermore, optimizations of greater complexity could be conducted, perhaps leveraging high-performance computing capability. In particular, advances in the integration of MBSE and simulation tools could be investigated, for instance ModelCenter can now directly generate simulation code from SysML diagrams, further improving traceability in architecture selection decisions [133].

An area with great potential for further analysis is the expansion of the application of rigorous uncertainty quantification. Further exploration of explicitly accounting for Type B as well as Type A uncertainty should be conducted. As highlighted in Case Study 3, uncertainty quantification and presentation to stakeholders can better inform a decision and highlight potential excursions to improve knowledge of the decision space. A robust exploration should be conducted of what measure of the objective such as mean, median, standard deviation, or something else, should be used for the selection decision, as well as how to integrate and interpret uncertainty in an optimization. In fact, some of the answers may be based on organizational preferences, regulations, or situational realities. Lastly, more analytical, as opposed to stochastic, methods to account for uncertainty could also be investigated.

A final area of investigation includes exploring validation of the methodology. Potential avenues to pursue this could include soliciting expert feedback, historical data analysis on program

or project performance and decision aid usage similar to this methodology, and direct analytical decomposition of the decision set up.

#### **6.4 Disclaimer**

The views expressed in this dissertation and the component case studies are solely the author's and do not represent the position of the U.S. Department of Defense, the U.S. Air Force, the Space and Missile Systems Center, Colorado State University or any other organization.

## REFERENCES

- [1] J. N. Martin, "Overview of an Emerging Standard on Architecture Evaluation," in *INCOSE International Symposium*, Adelaide, 2017.
- [2] United States Department of Defense, "Defense Acquisition Guidebook," 2017. [Online]. Available: <https://www.dau.mil/tools/dag>.
- [3] A. Qamar, S. J. I. Herzig, C. J. J. Paredis and M. Torngren, "Analyzing Semantic Relationships between Multiformalism Models for Inconsistency Management," in *2015 9th Annual IEEE International Systems Conference*, Vancouver, 2015.
- [4] C. Piaszczyk, "Model Based Systems Engineering with Department of Defense Architectural Framework," *Systems Engineering*, vol. 14, no. 3, pp. 305-326, 2011.
- [5] J. P. Lerat, "Three Reasons why Document-based SE (usually) works better than (most of) MBSE," in *INCOSE International Symposium*, Chicago, 2010.
- [6] Q. Do, S. Cook, T. Fosness and C. Haskins, "The Use of Models across the Contractual Boundary: Past, Present and Future," in *INCOSE International Symposium*, Las Vegas, 2014.
- [7] S. Pilemalm, N. Hallberg, M. Sparf and T. Niclason, "Practical Experiences of Model-Based Development: Case Studies from the Swedish Armed Forces," *Systems Engineering*, vol. 15, no. 4, pp. 407-421, 2012.
- [8] E. A. Bjorkman, S. Sarkani and T. A. Mazzuchi, "Using Model-Based Systems Engineering as a Framework for Improving Test and Evaluation Activities," *Systems Engineering*, vol. 16, no. 3, pp. 346-362, 2013.

- [9] M. Vasilevskaya, L. A. Gunawan, S. Nadjm-Tehrani and P. Herrmann, "Integrating security mechanisms into embedded systems by domain-specific modelling," *Security and Communication Networks*, vol. 7, no. 1, pp. 2815-2832, 2014.
- [10] S. Piggott, P. Melanson, L. Hartman and C. Adourian, "Advancing Model-Based Systems Engineering at a Small Space Agency," in *INCOSE International Symposium*, Utrecht, 2008.
- [11] Object Management Group, Inc, "MBSE Wiki," 2018. [Online]. Available: <http://www.omgwiki.org/MBSE/doku.php>. [Accessed 14 July 2018].
- [12] M. Bernal, J. R. Haymaker and C. Eastman, "On the role of computational support for designers in action," *Design Studies*, vol. 41, no. 1, pp. 163-182, 2015.
- [13] E. R. Carroll and R. J. Malins, "Systematic Literature Review: How is Model-Based Systems Engineering Justified?," United States Department of Energy, Albuquerque, 2016.
- [14] S. Friedenthal, A. Moore and R. Steiner, *A Practical Guide to SysML: The Systems Modeling Language*, San Mateo: Morgan Kaufmann, 2012.
- [15] S. Friedenthal and C. Oster, *Architecting Spacecraft With SysML: A Model-Based Systems Engineering Approach*, CA: CreateSpace Independent Publishing Platform, 2017.
- [16] J. Borky and T. Bradley, *Effective Model-Based Systems Engineering*, New York, New York: Springer, 2018.
- [17] A. Kossiakoff, W. N. Sweet, S. J. Seymore and S. M. Biemer, *Systems Engineering: Principles and Practice*, 2nd ed., Hoboken: John Wiley & Sons, Inc., 2011.
- [18] INCOSE, *Systems Engineering Handbook: A guide for system life cycle processes and activities*, San Diego, California: INCOSE, 2015.

- [19] A. Sharon, O. L. de Weck and D. Dori, "Improving Project-Product Lifecycle Management with Model-Based Design Structure Matrix: A Joint Project Management and Systems Engineering Approach," *Systems Engineering*, vol. 16, no. 4, pp. 413-426, 2013.
- [20] T. Bayer, S. Chung, B. Cole, B. Cooke, F. Dekens, C. Delp, I. Gontijo, K. Lewis, M. Moshir, R. Rasmussen and D. Wagner, "Early Formulation Model-Centric Engineering on NASA's Europa Mission Concept Study," in *INCOSE International Symposium*, Rome, 2012.
- [21] R. L. Keeney, "Decision Analysis: an Overview," *Operations Research*, vol. 30, no. 5, pp. 803-838, 1982.
- [22] K. Miettinen, *Nonlinear Multiobjective Optimization*, New York: Springer Science & Business Media, 1999.
- [23] R. T. Marler and J. S. Arora, "Survey of Multi-Objective Optimization Methods for Engineering," *Struct Multidisc Optim*, vol. 26, pp. 369-395, 2004.
- [24] W. R. Whittecar and M. P. Ferringer, "Global Coverage Constellation Design Exploration Using Evolutionary Algorithms," in *AIAA/AAS Astrodynamics Specialist Conference*, San Diego, 2014.
- [25] N. Raouf and S. H. Pourtakdoust, "Launch Vehicle Multi-Objective Reliability-Redundancy Optimization Using a Hybrid Genetic Algorithm-Particle Swarm Optimization," *Proc IMechE Part G: Journal of Aerospace Engineering*, vol. 229, no. 10, pp. 1785-1797, 2015.

- [26] V. Pandey and Z. P. Mourelatos, "A Distributed Pool Architecture for Highly Constrained Optimization Problems in Complex Systems Design," *Journal of Computing and Information Science in Engineering*, vol. 13, 2013.
- [27] N. F. Soria Zurita, M. K. Colby and I. Y. Tumer, "Design of Complex Engineered Systems Using Multi-Agent Coordination," *Journal of Computing and Information Science in Engineering*, vol. 18, 2018.
- [28] P. M. Zadeh and M. S. Shirazi, "Multidisciplinary Design Optimization Architecture to Concurrent Design of Satellite Systems," *Proc IMechE Part G: Journal of Aerospace Engineering*, vol. 231, no. 10, pp. 1898-1916, 2017.
- [29] I. Min, E. Mahr, R. Vaughan and I. Kellogg, "Rapid Space Architecture Development and Analysis Using an Integrated Toolset," in *IEEE Aerospace Conference Proceedings*, 2004.
- [30] S. J. I. Herzig, S. Mandutianu, H. Kim, S. Hernandez and T. Imken, "Model-Transformation-Based Computational Design Synthesis for Mission Architecture Optimization," in *IEEE Aerospace Conference*, 2017.
- [31] T. Mosher, "Spacecraft Design Using a Genetic Algorithm," in *IEEE Aerospace Conference*, 1998.
- [32] S. W. Mitchell, "Efficient Management of Configurations in the Model Based System Development of a Common Submarine Combat System," in *GMU-AFCEA Symposium*, Fairfax, 2011.
- [33] J. Ryan, S. Sarkani and T. Mazzuchi, "Leveraging Variability Modeling Techniques for Architecture Trade Studies and Analysis," *Systems Engineering*, vol. 17, no. 1, pp. 10-25, 2014.

- [34] I. Min and R. Noguchi, "The Architecture Design and Evaluation Process: A Decision Support Framework for Conducting and Evaluating Architecture Studies," The Aerospace Corporation, El Segundo, 2016.
- [35] R. D. Coder and M. J. Holzinger, "Multi-Objective Design of Optical Systems for Space Situational Awareness," *Acta Astronautica*, vol. 128, pp. 669-684, 2016.
- [36] D. Selva, B. Cameron and E. F. Crawley, "A Rule-Based Method for Scalable and Traceable Evaluation of System Architectures," *Res Eng Design*, p. 325–349, 2014.
- [37] M. W. Maier and E. Rechtin, *The Art of Systems Architecting*, 2nd ed., Boca Raton: CRC Press LLC, 2000.
- [38] A. Fisher, M. Nolan, S. Friedenthal, M. Loeffler, M. Sampson, M. Bajaj, L. VanZandt, K. Hovey, J. Palmer and L. Hart, "Model Lifecycle Management for MBSE," in *INCOSE International Symposium*, Las Vegas, 2014.
- [39] M. Blackburn, M. Bone and G. Witus, "Transforming System Engineering Through Model-Centric Engineering," Stevens Institute of Technology, New Jersey, 2015.
- [40] C. Paredis and T. Johnson, "Using OMG's SysML to Support Simulation," in *Simulation Conference, WSC 2008*, 2008.
- [41] S. C. Spangelo, D. Kaslow, C. Delp, B. Cole, L. Andersen, E. Fosse, B. S. Gilber, L. Hartman, T. K. Kahn and J. Cutler, "Applying Model Based Systems Engineering (MBSE) to a Standard Cube-Sat," in *IEEE Aerospace Conference*, 2012.
- [42] R. Kawahara, D. Dotan, T. Sakairi, K. Ono, H. Nakamura, A. Kirshin, S. Hirose and H. Ishikawa, "Verification of Embedded System's Specification Using Collaborative



- Simulation of SysML and Simulink Models," in *International Conference on Model-Based Systems Engineering*, Haifa, 2009.
- [43] D. Spyropoulos and J. S. Baras, "Extending Design Capabilities of SysML with Trade-off Analysis: Electrical Microgrid Case Study," *Procedia Computer Science*, vol. 16, pp. 108-117, 2013.
- [44] P. Couturier, M. Lo, A. Imoussaten, V. Chapurlat and J. Montmain, "Tracking the Consequences of Design Decisions in Mechatronic Systems Engineering," *Mechatronics*, vol. 24, pp. 763-774, 2014.
- [45] J. T. Newcomer, "A New Approach to Quantification of Margins and Uncertainties for Physical Simulation Data," United States Department of Energy, Albuquerque, 2012.
- [46] H. Markowitz, "Portfolio Selection," *The Journal of Finance*, vol. 7, no. 1, pp. 77-91, 1952.
- [47] N. Davendralingam and D. A. DeLaurentis, "A Robust Portfolio Optimization Approach to System of System Architectures," *Systems Engineering*, vol. 18, no. 3, pp. 269-283, 2015.
- [48] R. A. McDonald, "Error Propagation and Metamodeling for a Fidelity Tradeoff Capability in Complex Systems Design," Georgia Institute of Technology, 2006.
- [49] B. S. Blanchard and W. J. Fabrycky, *Systems Engineering and Analysis*, 4th ed., Upper Saddle River, New Jersey: Pearson Prentice Hall, 2006.
- [50] M. Xu, J. Wang, A. Zhang and S. Liu, "Probabilistic Value-Centric Optimization Design for Fractionated Spacecrafts Based on Unscented Transformation," *Mathematical Problems in Engineering*, vol. 2013, pp. 1-10, 2013.

- [51] A. Jafarsalehi, H. R. Fazeley and M. Mirshams, "Conceptual Remote Sensing Satellite Design Optimization Under Uncertainty," *Aerospace Science and Technology*, vol. 55, pp. 377-391, 2016.
- [52] D. W. Hubbard and R. Seiersen, *How to Measure Anything in Cybersecurity Risk*, Hoboken: John Wiley & Sons, Inc., 2016.
- [53] D. Kahneman and A. Tversky, "Subjective Probability: A Judgement of Representativeness," *Cognitive Psychology*, vol. 3, pp. 430-454, 1972.
- [54] M. Lewis, "How Two Trailblazing Psychologists Turned the World of Decision Science Upside Down," 14 November 2016. [Online]. Available: <https://www.vanityfair.com/news/2016/11/decision-science-daniel-kahneman-amos-tversky>. [Accessed 12 August 2018].
- [55] R. Howard, Interviewee, *Harvard Business Review Podcast*. [Interview]. 20 November 2014.
- [56] P. E. Meehl, *Clinical versus Statistical Prediction; A Theoretical Analysis and a Review of the Evidence*, Minneapolis: University of Minnesota Press, 1954.
- [57] P. E. Meehl, "Causes and Effects of My Disturbing Little Book," *Journal of Personality Assessment*, vol. 50, pp. 370-375, 1986.
- [58] W. Perry, D. Signori and J. Boon, *Exploring Information Superiority: A Methodology for Measuring the Quality of Information and Its Impact on Shared Awareness*, Santa Monica: RAND Corporation, 2004.

- [59] ISO/IEC 19514:2017, Information Technology-Object Management Group Systems Modeling Language (OMG SysML), Int Organ Standardization/Int Electrotechnical Comm May 2017.
- [60] Joint Chiefs of Staff, "Chairman of the Joint Chiefs of Staff Instruction 3170.01, Joint Capabilities Integration and Development System (JCIDS)," Washington, DC, 2015.
- [61] J. R. Turner, The Handbook of Project Based Management, 3rd ed., New York City: McGraw-Hill, 2009.
- [62] W. D. Larson and J. R. Wertz, Space Mission Analysis and Design, 3rd ed., Torrance, California: Microcosm Press, 1999.
- [63] United States Department of Defense, "Department of Defense Instruction 5000.02, Operation of the Defense Acquisition System," Washington, DC, 2015.
- [64] United States Air Force Cost Analysis Agency, "Cost Risk and Uncertainty Handbook," US Air Force, 2007.
- [65] E. Mahr and S. Holloman, "Why Programs Overrun," The Aerospace Corporation, 2016.
- [66] Office of the Deputy Undersecretary of Defense for Acquisition, Technology, and Logistics, "Systems Engineering Guide for Systems of Systems," Washington, DC, 2008.
- [67] Office of the Assistant Secretary of the Air Force (Acquisition), "Monthly Acquisition Report (MAR) Guidebook," 12 October 2016. [Online]. Available: <https://www.dau.mil/cop/bes/DAU%20Sponsored%20Documents/Monthly%20Acquisition%20Report%20MAR%20Guidebook.pdf>. [Accessed 17 March 2018].

- [68] AcqNotes, "Selected Acquisition Report (SAR)," 11 July 2017. [Online]. Available: <http://acqnotes.com/acqnote/acquisitions/selected-acquisition-report-sar>. [Accessed 17 March 2018].
- [69] Joint Chiefs of Staff, "Manual for the Operation of the Joint Capabilities Integration and Development System," Washington, DC, 2015.
- [70] AcqNotes, "Net-Ready Key Performance Parameter (NR-KPP)," 12 July 2017. [Online]. Available: <http://acqnotes.com/acqnote/acquisitions/net-ready-key-performance-parameter-nr-kpp>. [Accessed 19 March 2018].
- [71] Joint Chiefs of Staff, "Chairman of the Joint Chiefs of Staff Instruction 6212.01F, Net Ready Key Performance Parameter," Washington, DC, 2012.
- [72] Director of Operational Test and Evaluation, "The Value of Operational Test and Evaluation," 15 May 2014. [Online]. Available: [http://www.dote.osd.mil/pub/presentations/Value\\_of\\_Operational\\_Testing\\_and\\_Evaluation.pdf](http://www.dote.osd.mil/pub/presentations/Value_of_Operational_Testing_and_Evaluation.pdf). [Accessed 17 March 2018].
- [73] R. E. Thompson, J. M. Colombi, J. Black and B. J. Ayres, "Disaggregated Space System Concept Optimization: Model-Based Conceptual Design Methods," *Systems Engineering*, vol. 18, no. 6, pp. 549-567, 2015.
- [74] Phoenix Integration, "ModelCenter Integrate," 2018. [Online]. Available: <https://www.phoenix-int.com/product/modelcenter-integrate/>. [Accessed 26 March 2018].
- [75] Microsoft, "Excel 2016," 2018. [Online]. Available: <https://products.office.com/en-us/excel>. [Accessed 28 March 201].

- [76] The MathWorks, Inc., "MATLAB," 2018. [Online]. Available: <https://www.mathworks.com/products/matlab.html>. [Accessed 28 March 2018].
- [77] Analytical Graphics, Inc, "Engineering Tools," 2018. [Online]. Available: <https://www.agi.com/products/engineering-tools>. [Accessed 27 March 2018].
- [78] IBM, "Rational Rhapsody," 2018. [Online]. Available: <https://www.ibm.com/us-en/marketplace/rational-rhapsody>. [Accessed 27 March 2018].
- [79] Sparx System Pty Ltd., "Enterprise Architect," 2018. [Online]. Available: <http://sparxsystems.com/products/ea/>. [Accessed 27 March 2018].
- [80] D. P. Loucks and E. van Beek, Water Resource Systems Planning and Management, An Introduction to Models, Methods and Applications, Turin: United Nations Educational, Scientific, and Cultural Organization, 2005.
- [81] A. R. DeWispelare, H. L. T, E. J. Bonano and R. T. Clemen, "Background Report on the Use and Elicitation of Expert Judgement," San Antonio, 1994.
- [82] S. Lichtenstein, B. Fischhoff and L. D. Philips, "Calibration of Probabilities: The State of the Art to 1980," in *Judgement under Uncertainty: Heuristics and Biases*, D. Kahneman, P. Slovic and A. Tversky, Eds., Cambridge, Cambridge University Press, 1982.
- [83] M. Jamshidi, Ed., System of Systems Engineering, Hoboken, NJ: Wiley, 2009.
- [84] Air Force Space Command Public Affairs, "Hyten announces Space Enterprise Vision," 2016. [Online]. Available: <http://www.af.mil/News/ArticleDisplay/tabid/223/Article/719941/hyten-announces-space-enterprise-vision.aspx>.

- [85] Air Force Space Command Public Affairs, "AFSPC commander unveils three major space initiatives at 33rd Space Symposium," 2017. [Online]. Available: <http://www.24af.af.mil/News/Article-Display/Article/1145448/afspc-commander-unveils-three-major-space-initiatives-at-33rd-space-symposium/>.
- [86] E. Pawlikowski, D. Loverro and T. Cristler, "Space Disruptive Challenges, New Opportunities, and New Strategies," *Strategic Studies Quarterly*, vol. 6, no. 1, pp. 27-54, 2012.
- [87] United States Department of Defense, "DoD Dictionary of Military and Associated Terms," Washington, D.C., 2017.
- [88] C. Clark, "JICSPOC morphs to 'National Space Defense Center;' what it means," 2017. [Online]. Available: <http://breakingdefense.com/2017/04/jicspoc-morphs-to-national-space-defense-center-what-it-means/>. [Accessed 30 April 2017].
- [89] Office of the Deputy Undersecretary of Defense for Acquisition, Technology, and Logistics, "Better Buying Power 3.0," 9 April 2015. [Online]. Available: [http://www.acq.osd.mil/fo/docs/betterBuyingPower3.0\(9Apr15\).pdf](http://www.acq.osd.mil/fo/docs/betterBuyingPower3.0(9Apr15).pdf). [Accessed 30 April 2017].
- [90] W. Crossley and E. Williams, "Simulated Annealing and Genetic Algorithm Approaches for Discontinuous Coverage Satellite Constellation Design," *Engineering Optimization*, vol. 32, no. 3, pp. 353-371, 2007.
- [91] E. Burgess, "Modeling R&D Budget Profiles," in *The Society of Cost Estimating and Analysis (SCEA) and International Society of Parametric Analysts (ISPA) Joint Annual Conference Proceedings*, Orlando, 2012.

- [92] United States Air Force, "Cost Estimating Relationship for Military Communications Satellites, Unmanned Satellite Cost Model (USCM)," United States Air Force, Los Angeles, 2014.
- [93] Office of the Deputy Undersecretary of Defense for Acquisitions, Technology, and Logistics, "Enhancing Competition Awareness in DoD," 2010. [Online]. Available: [www.acq.osd.mil/dpap/cpic/cp/docs/training.ppt](http://www.acq.osd.mil/dpap/cpic/cp/docs/training.ppt). [Accessed 18 March 2017].
- [94] M. King and M. Riccio, "Military Satellite Communications: Then and Now," *Crosslink Magazine*, 2010.
- [95] United States Government Accountability Office, "Additional Knowledge Would Better Support Decisions About Disaggregating Large Satellites," Washington, D.C., 2014.
- [96] J. Gross and S. Rudolph, "Rule-Based Spacecraft Design Space Exploration and Sensitivity Analysis," *Aerospace Science and Technology*, no. 59, pp. 162-171, 2016.
- [97] Headquarters Air Force Office of Aerospace Studies, "Analysis of Alternatives Handbook," U.S. Air Force, 2016.
- [98] J. Martin, *personal correspondence*, 2017.
- [99] J. G. Walker, "Satellite Constellations," *Journal of the British Interplanetary Society*, vol. 37, pp. 559-571, 1984.
- [100] Z. Stojanovski and D. Selva, "Optimizing Commonality and Performance in Platform-Based Earth Observing SmallSat Architectures," in *2017 IEEE Aerospace Conference*, Big Sky, 2017.
- [101] C. Y. Tsai, C. J. Chen and Y. T. Lo, "A Cost-Based Module Mining Method for Assemble-to-Order Strategy," *Journal of Intelligent Manufacturing*, vol. 25, pp. 1377-1392, 2014.

- [102] National Institute of Standards and Technology, "Evaluating Uncertainty Components: Type A," [Online]. Available: <https://physics.nist.gov/cuu/Uncertainty/typea.html>. [Accessed 25 June 2018].
- [103] National Institute of Standards and Technology, "Evaluating Uncertainty Components: Type B," [Online]. Available: <https://physics.nist.gov/cuu/Uncertainty/typeb.html>. [Accessed 25 June 2018].
- [104] T. Bradley, *ENGR 513 Lecture 2.17.16*, Fort Collins: Colorado State University, 2016.
- [105] A. Greenberg, "The Untold Story of NotPetya, The Most Devastating Cyberattack in History," 22 August 2018. [Online]. Available: <https://www.wired.com/story/notpetya-cyberattack-ukraine-russia-code-crashed-the-world/>. [Accessed 8 September 2018].
- [106] The White House, "National Security Presidential Directive/NSPD-54," 8 January 2008. [Online]. Available: <https://fas.org/irp/offdocs/nspd/nspd-54.pdf>. [Accessed 28 July 2018].
- [107] S. Northcutt, "The Attack Surface Problem," 7 January 2011. [Online]. Available: [www.sans.edu/research/security-laboratory/article/did-attack-surface](http://www.sans.edu/research/security-laboratory/article/did-attack-surface). [Accessed 28 July 2018].
- [108] J. M. Borky, *ENGR 567 Lecture, Session 8: Service-Oriented Architecture Part 1*, Colorado Springs: Colorado State University, 2016.
- [109] D. A. Deptula, "Evolving Technologies and Warfare in the 21st Century: Introducing the "Combat Cloud"," Mitchell Institute for Aerospace Studies, 2016.
- [110] K. Stroker, *Microsoft IoT Hackster LIVE 2017: Beyond the Device*, Microsoft, 2017.



- [111] C. Phan, L. Lee and R. Fleischer, "Service Oriented Architecture (SOA) - Security Challenges and Mitigation Strategies," in *Military Communications Conference, 2007*, Orlando, 2007.
- [112] W. Larson, Ed., *Space Mission Analysis and Design CORE*, Colorado Springs: CEI Publications, 2015.
- [113] Sandia National Laboratories, "Dakota, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 6.7 User's Manual," Sandia National Laboratories, Albuquerque, 2017.
- [114] Z. Li and J. Keung, "Software Cost Estimation Framework for Service-Oriented Architecture Systems using Divide-and-Conquer Approach," in *2010 Fifth IEEE International Symposium on Service Oriented System Engineering*, Nanjing, 2010.
- [115] B. W. Boehm, C. Abts, A. W. Brown, S. Chulani, B. K. Clark, E. Horowitz, R. Madachy, D. J. Reifer and B. Steece, *Software Cost Estimation with COCOMO II*, New Jersey: Prentice Hall PTR, 2000.
- [116] CSP Associates, "GFSC Ground Systems Cost Model Study; Final Results: Phases I, II, and III," Cambridge, 1999.
- [117] J. Carraway, "Lowering Operations Costs Through Complexity Metrics," in *Space Ops '94*, Greenbelt, 1994.
- [118] J. M. Borky, "Cybersecurity For Systems Engineers," Colorado Springs, 2018.
- [119] National Institute of Standards and Technology, "Guide for Conducting Risk Assessments," U.S. Department of Commerce, Gaithersburg, 2012.

- [120] S. Gandel, "Lloyd's CEO: Cyber Attacks Cost Companies \$400 Billion Every Year," 23 January 2015. [Online]. Available: <http://fortune.com/2015/01/23/cyber-attack-insurance-lloyds/>. [Accessed 15 July 2018].
- [121] A. Mukhopadyay, S. Chatterjee, K. K. Bagchi, P. J. Kirs and G. K. Shukla, "Cyber Risk Assessment and Mitigation (CRAM) Framework Using Logit and Probit Models for Cyber Insurance," *Information Systems Frontiers*, pp. 1-22, 2017.
- [122] C. Martin and K. A. Abuosba, "Utilizing a Service Oriented Architecture for Information Security Evaluation and Quantification," in *2007 2nd IEEE/IFIP International Workshop on Business-Driven IT Management*, Munich, 2007.
- [123] R. A. Howard and A. E. Abbas, *Foundations of Decision Analysis*, New York: Prentice Hall, 2015.
- [124] Verizon, "2018 Data Breach Investigations Report," Verizon, 2018.
- [125] D. W. Hubbard and R. Seiersen, "How to Measure Anything in Cybersecurity Risk," 2018. [Online]. Available: <https://www.howtomeasureanything.com/cybersecurity/>. [Accessed 29 August 2018].
- [126] National Institute of Standards and Technology, "Risk Management," 21 August 2018. [Online]. Available: [https://csrc.nist.gov/projects/risk-management/risk-management-framework-\(RMF\)-Overview](https://csrc.nist.gov/projects/risk-management/risk-management-framework-(RMF)-Overview). [Accessed 10 September 2018].
- [127] A. Stuart and K. Ord, *Kendall's Advanced Theory of Statistics: Volume I Distribution Theory*, London: Edward Arnold Publishers Ltd., 1994.

- [128] Center for Software Engineering at the University of Southern California, "COCOMO II Model Definition Manual Version 2.1," University of Southern California, Los Angeles, 2000.
- [129] D. Ingold, B. Boehm, S. Koolmanojwong and J. A. Lane, "A Model for Estimating Agile Project Schedule Acceleration," University of Southern California, Los Angeles, 2012.
- [130] F. P. Brooks, *The Mythical Man-Month*, Crawfordsville: Addison-Wesley Longman, Inc., 1995.
- [131] L. A. J. Cox, "What's Wrong with Risk Matrices," *Risk Analysis*, vol. 28, no. 2, pp. 497-512, 2008.
- [132] D. Hubbard and D. Evans, "Problems with Scoring Methods and Ordinal Scales in Risk Assessment," *IBM Journal of Research and Development*, vol. 54, no. 3, 2010.
- [133] Phoenix Integration, "MBSEpak," 2018. [Online]. Available: <https://www.phoenix-int.com/product/mbsepak/>. [Accessed 1 October 2018].
- [134] M. LaSorda, J. M. Borky and R. M. Sega, "Model-Based Architecture and Programmatic Optimization for Satellite System-of-Systems Architectures," *Systems Engineering*, vol. 21, no. 4, pp. 372-387, 2018.
- [135] M. LaSorda, J. Borky and R. Sega, "Model-Based Architecture Optimization for Major Acquisition Analysis of Alternatives," in *2018 IEEE Aerospace Conference*, Big Sky, 2018.